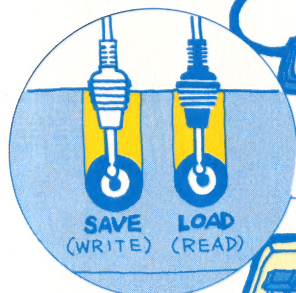
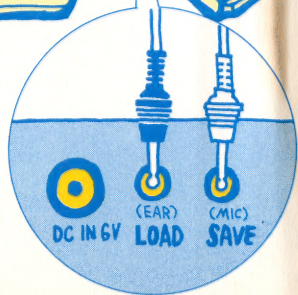
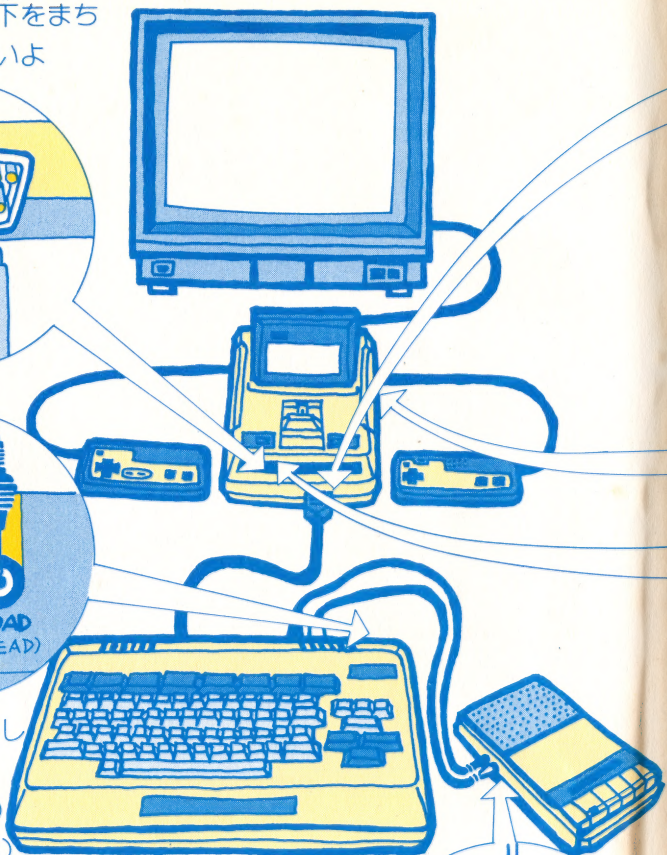
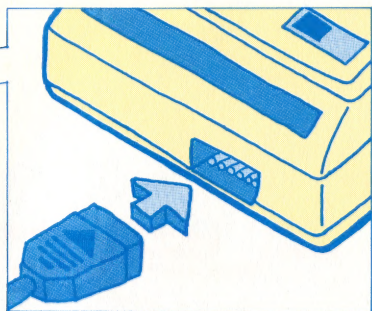
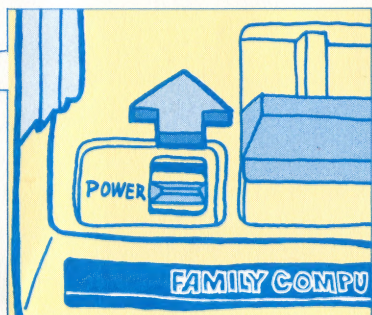
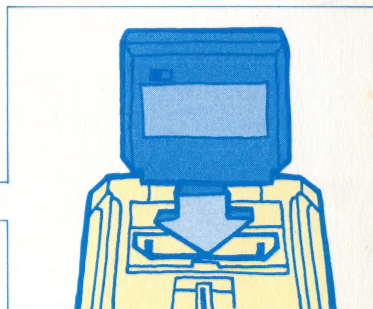
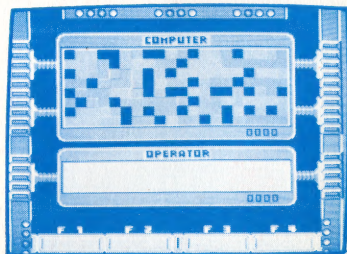
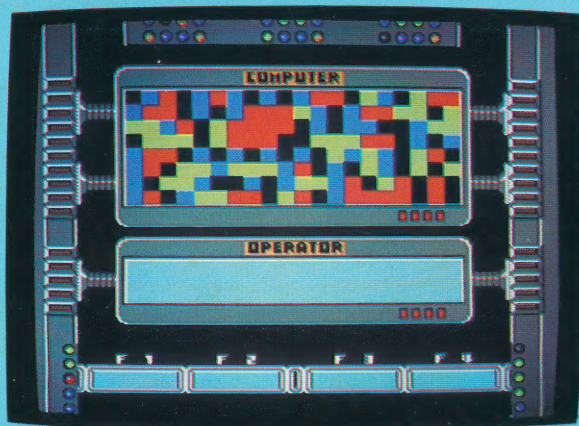
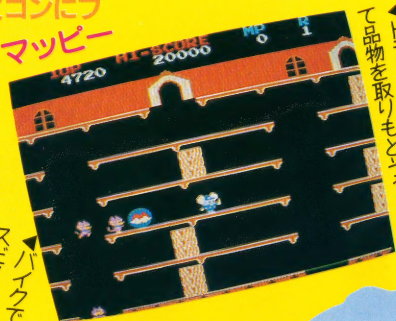
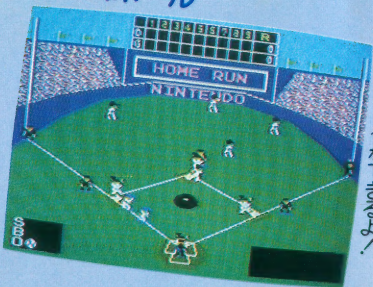
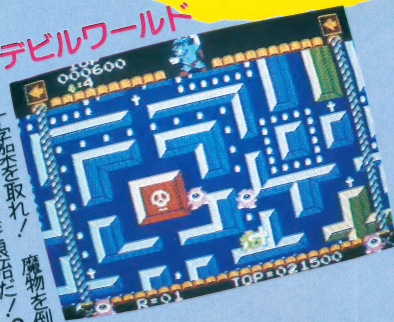
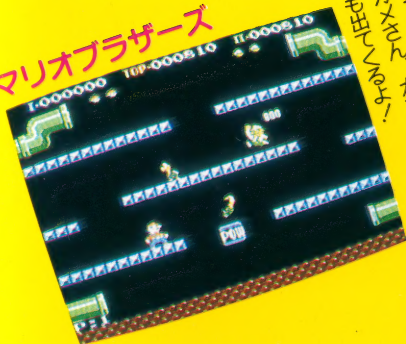
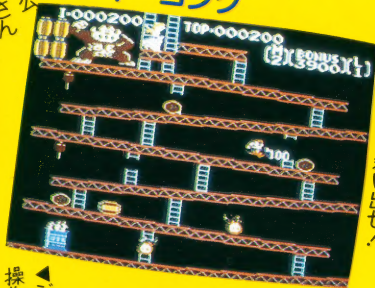
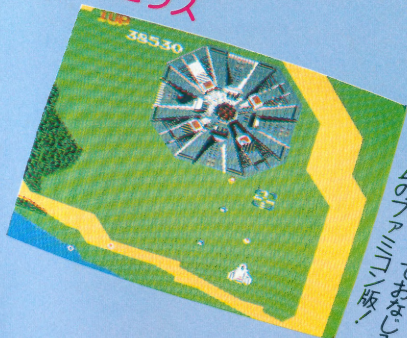
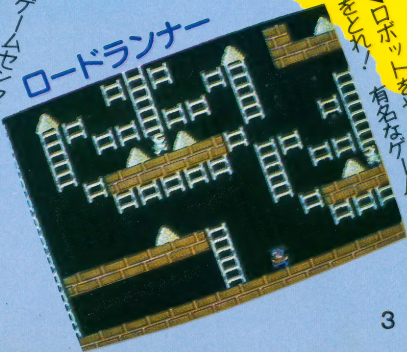
任天堂の
ファミリーベーシック™入門
すぐに遊べる
ゲームプログラム
10本入り
オリジナル
ゲーム
が作れるよう
になる!
FAMILY BASIC
V3でも
使える
テクノポリス編集部：編著
ファミリーベーシックは任天堂の商標です。
FAMILY BASIC
FAMILY COMPUTER
Nintendo

ファミリーベーシ
▼コネクタは上下をまちがえると入らないよ
SAVE
(WRITE)
LOAD
(READ)
▲ケーブルを逆にさし込まないようにね
SAVE(WRITE)
→SAVE(MIC)
LOAD(READ)
→LOAD(EAR)
▶ラジカセでは、SAVEはMIC(マイクロホン)、LOADはEAR(イヤホン)につなぐ
DC IN 6V
(EAR)
LOAD
(MIC)
SAVE

ックのつなぎ方

①ファミリーコンピュータとキーボードのコネクタをつなぐ

②ファミリーベーシックのカセットをファミコンにさす

③ファミリーコンピュータの電源スイッチを入れる

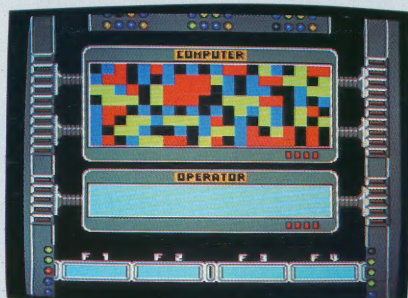
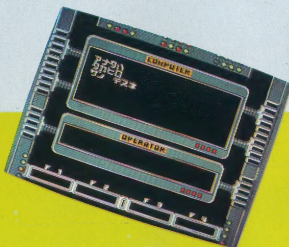
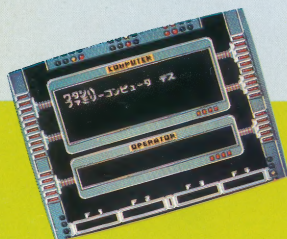
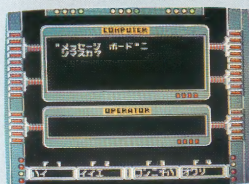
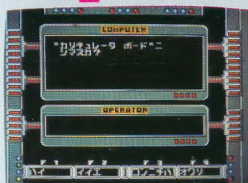
④画面が現れてファミリーベーシックのスタートだ！

任天堂の

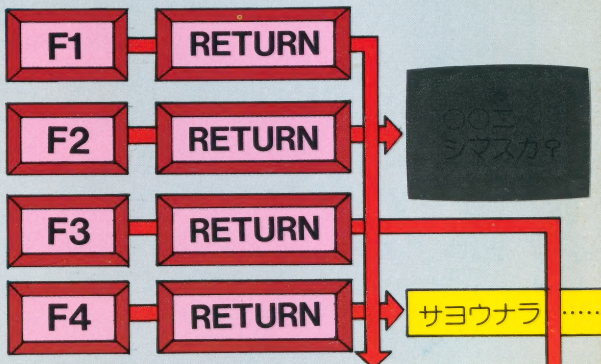
ファミリーベーシック™入門

ファミコンにはいろんなゲームがあるけれど……

ファミリーベーシックは、ここにあげたようなファミコンのゲーム・カセットとはちょっとちがうね。キーボードがついてるし、カセットも大きくてしかもスイッチがついていたりする。外見だけじゃなくて、遊び方もぜんぜんちがうんだよ。カセットゲームみたいなのを自分で作っちゃおう、ほかにもいろんなことをしてみよう、そうやって楽しもう、というのがファミリーベーシックなんだ。ゲームもいいけどファミリーベーシックも楽しいよ。さあファミコンにファミリーベーシックをつなごう！

マリオブラザーズ
▶マリオが主人公
カメさん、カニさん
も出てくるよ！
ドンキーコング
▶ドンキーコングがさらってい
ったレディを救い出せ！
ドンキーコングJr.
▶ジュニアを
操作してドン
キーコングを助け出そう！
F-1レース
▶競争相手のマシンをかわせ！
ゴールめざしてまっしぐら
ゼビウス
▶ゲームセンターでおなじみ
のゲームのファミコン版！
ロードランナー
▶ロボットをやっつけて金塊
をとれ！　有名なゲームだよ

ファミリーベーシックのスイッチを入れてみよう!

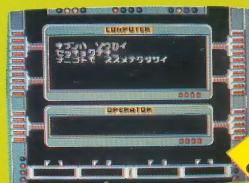
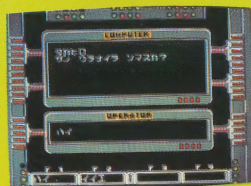
ファミコンにファミリーベーシックのカセットを入れて、キーボードをつないで、スイッチオン！　ファミリーベーシックの不思議で楽しい世界がはじまるよ。キーボードにいっぱいならんでるボタンをちょっと押しただけで、ファミリーコンピュータがあいさつをしてくれるんだ。

これからはキミがファミコンのご主人さま。ファミコンがキミの名前をおぼえて、ファミリーベーシックのなかにある5つの世界へキミを案内してくれるよ。キミはなにからはじめるのかな？

キミのいうとおりにファミコンが動く

▼ファミコンがキミの名前をきいてきたら、キーボードで名前を打ってみよう。押されたキーに書いてある文字が、画面のわくのなかに1つずつ出てくるね。ローマ字で入れるのならそのままでいいけれど、カタカナのときは カナ キーを1回押してからだよ。

ファミリーベーシックのメニュー キミはどれを選ぶ?

ファミコンが4つの世界に誘いかけてくる。そこで遊びたかったら、F1キー。いやならF2キーを押せばいいけど、あんまり長いあいだ何もしていないと時間ぎれで終了になっちゃうよ。F4キーなら、ファミコンとお別れだ。

プログラムを作って遊ぶ
★GAME BASIC
アッというまに計算してくれる
★カリキュレータボード
ファミコンが伝言板になるよ
★メッセージボード
コンピュータの音楽演奏だ
★ミュージックボード

F3キーを押すとあいさつしてからコンピュータが占いをしてくれるよ。これがあたるんだ。

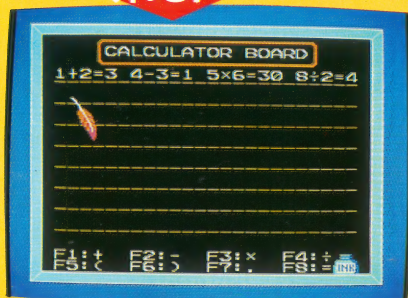
ファミコンがうらないをしてくれるよ!

コンニチハ!

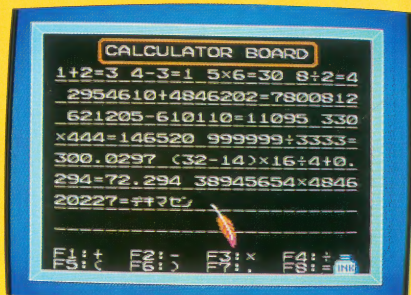
計算、伝言、ミュージック、そして……

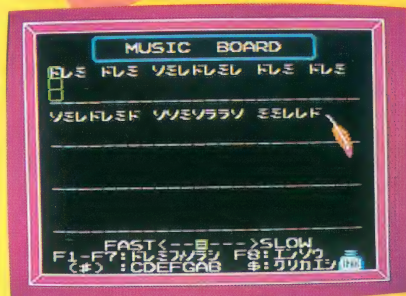
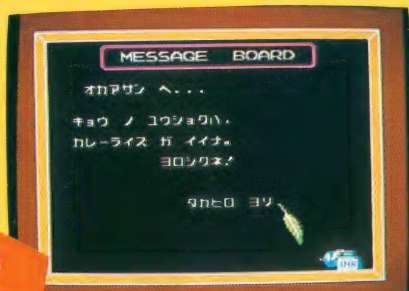
はじめてでもかんたんに遊べるのが、"カリキュレータボード"、"メッセージボード"、"ミュージックボード" だね。ファミコンは、この3つをまとめて "カクシュボード" と呼ぶこともあるよ。

この3つのボードでの主役は羽根ペンとインクツボ。計算・伝言・ミュージックの遊びが楽しくできるようになってるよ。

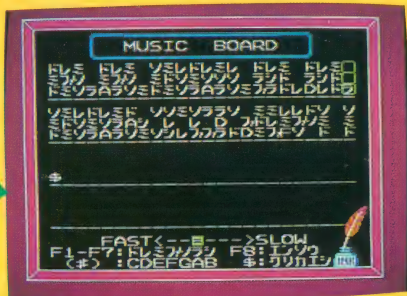
カリキュレータボード

すごいすごい！
どんどん計算してくれる！

カリキュレータボードでは、キミが計算式を打ちこんで "="(イコール) を入れるだけで、コンピュータがすばやく計算してくれるよ。算数の宿題もやってくれそうだね。小数点の計算だってやってくれるよ！

CALCULATOR BOARD
1+2=3 4-3=1 5×6=30 8÷2=4
2954610+4846202=7800812
621205-610110=11095 330
×444=146520 999999÷3333=
300.0297 (32-14)×16÷4+0.
294=72.294 38945654×4846
20227=デキマセン
F1:+ F2:- F3:× F4:÷
F5:(F6:) F7:. F8:=
INK

メッセージボード

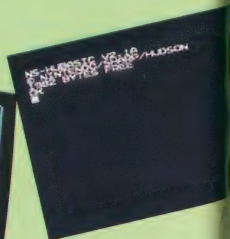
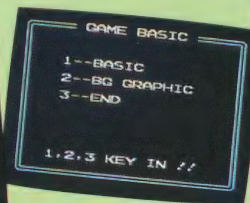
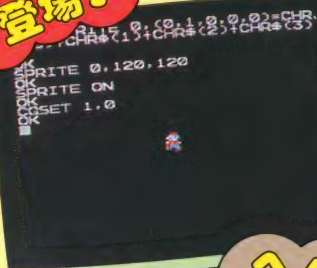
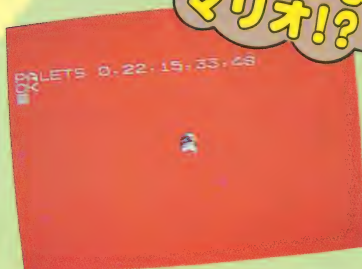
友だちへの伝言、ファミコンに伝えてもらおう

なんとなく口でいいにくいことを友だちやおかあさんに伝えたいとき、メッセージボードが役に立つよ。友だちと今度遊ぶ日を約束したり、おかあさんに夕食をカレーにしてねと頼んだり、いろいろ使えるね。

ミュージックボード

なんと3つの音を同時に出して、和音までこなすのがミュージックボード。ドレミ……を画面に書きこめば、そのとおりに自動演奏してくれるよ。知ってる曲をアレンジするのも楽しいね。

ベーシックならこんなにいろんなことができるんだ

GAME BASICはほかのボードとちょっとちがうね。ベーシック（BASIC）というのは、コンピュータの言葉のひとつなんだ。この言葉を使って、ファミコンに命令すれば、画面にいろんなキャラクタが出てきたり、動いたり、ゲームができたりするんだぞ。

この本は、このベーシックをキミが使いこなせるようにベーシックの言葉の意味や使い方を紹介してるんだ。

ここにのってる画面は、みんなこの本でやってることだよ。やり方はあとのほうで説明するけど、まずはどんなことができるのか見ておこう。

アキレスが
パタパタ

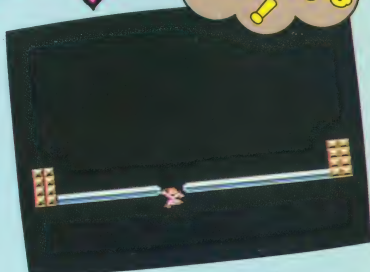
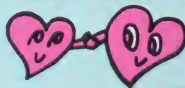
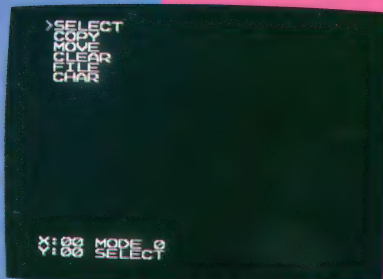
もしもし
カメさん♪

敵が追っか
けてくる!!

マリオと
レディの出会い

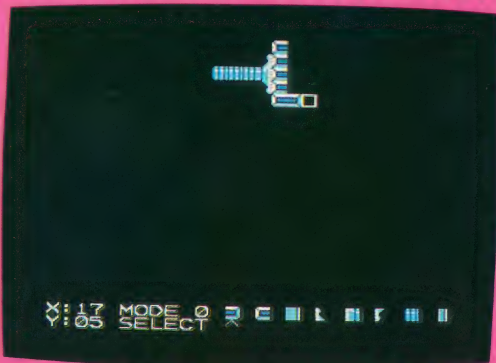
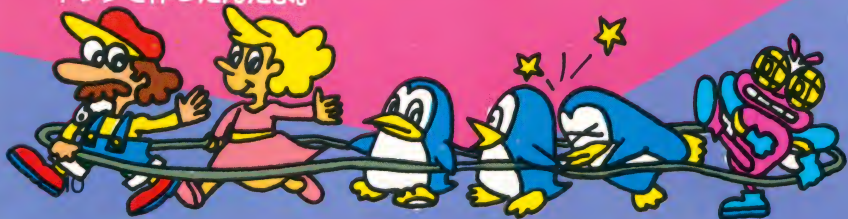
レディが
リンゴを
取るよ

SCORE= 20

キャッ!!
おとしあな
!

BG-GRAPHICで背景の絵がかけちゃう!

GAME BASICにはもう1つ、2のキーで選ぶBG-グラフィック(GRAPHIC)というのがあるね。

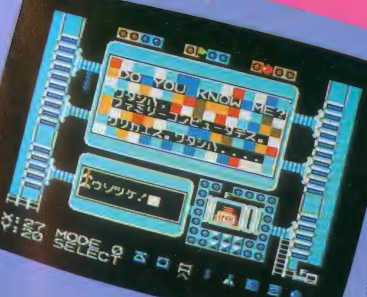
これはファミリーベーシックでオリジナル・ゲームを作って遊ぶときに、かっこいい背景を作るためのものなんだ。はじめは絵をかくだけでも楽しいよ。

BG-グラフィックの使い方は、この本の6章でもくわしく説明しているよ。といっても、図形や色を選んで、キーボードを押していくだけだから、ちょっとやればすぐ覚えられそう。

ここで使える図形は、なんと104種類。しかも、それぞれに4通りの色の組み合わせがあるんだから、キミのアイデア次第で楽しい背景がどんどんできちゃう。右のページの絵は全部BG-グラフィックで作ったんだよ。

▲図形を選んでキーを押していくと、ほらこんな絵が……

インクつぼだって
ごらんのとおり

あとでゲームに
使おうかな

ファミコンの
マネしちゃった！

床もいろんなのが
かけるんだね

キミだけのオリジナルゲームをいっぱい作ろう

ベーシックとBG - グラフィックの使い方を覚えればキミにも楽しいオリジナル・ゲームが作れるよ。9章のゲーム・プログラム集も、もちろん、ベーシックとBG - GRAPHICだけで作ってあるんだ。

ここにある画面が、9章にのってるゲームだよ。なんだかすぐにでも遊びたくなっちゃうほど、楽しそうだね。

9章のプログラムをファミリーベーシックに入れれば遊べるよ。

ファイティング・レディ

▲いじわるなファイターフライをレディがやっつけるよ！

不思議の森のピクニック

▲なんだか目が回っちゃう不思議なゲーム

ハエ・ハエ・カカカ！

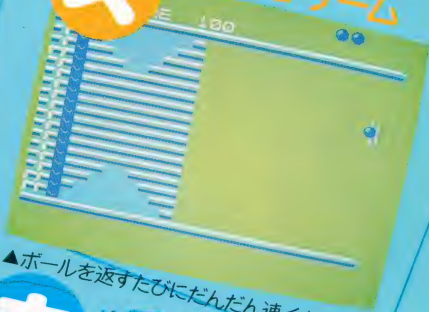
▲ハエの大群がマリオにおそいかかった！

バウンド・ボール

▲ボールがはねまわるよ

▲スターシップに乗りこんだキミは、おそろしい敵・スピナーと戦いつづけるのだ

▲変なブルドーザーで鉱石をひろい集めよう

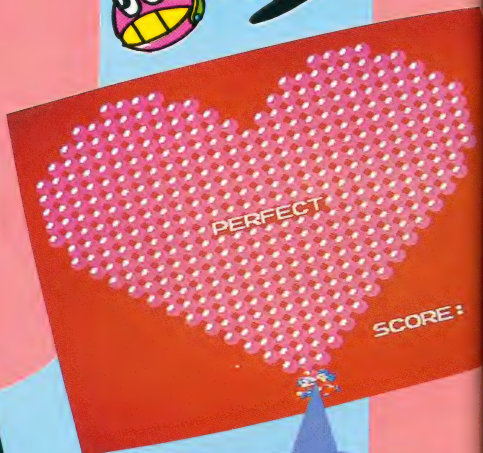
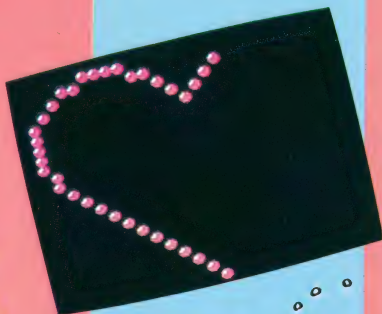
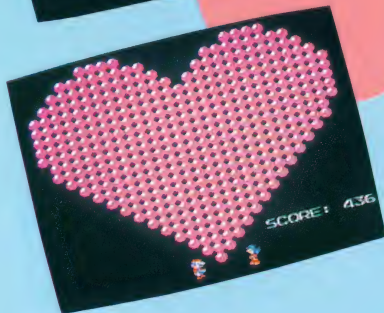
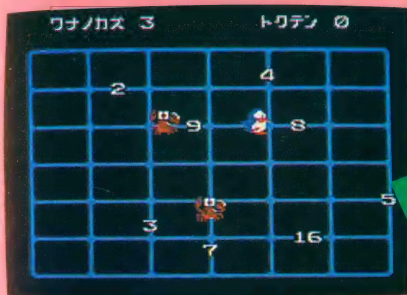
▲ニットピッカーにぶつからないように、ファイターフライを飛ばせてあげて！

▲ボールを返すたびにだんだん速くなるよ！

▲上空からおそいかかってくるニタニタ軍団をうちおとせ！

▲うまく旗やリンゴがとれるかな

ファミリーベーシックV3ってどんなことができるの？

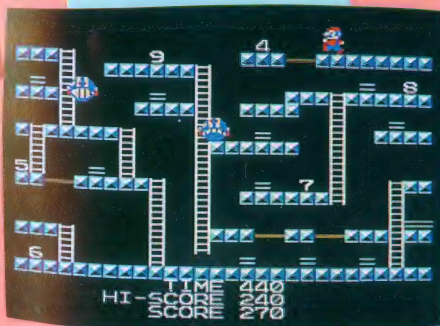
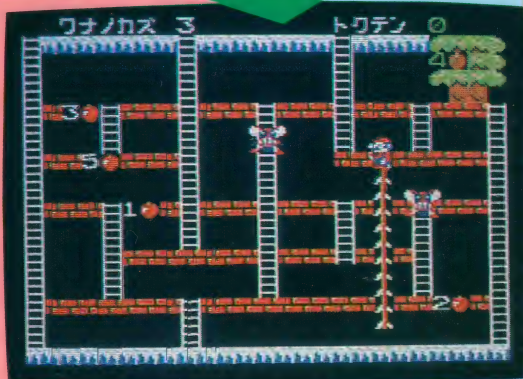
新しい仲間ファミリーベーシックV3は4つのゲームが入っていて、ベーシックも使いやすくなったぞ。

マイクであそぼうGAME0
ハート

マリオとレディのラブシーンで完成！

こわいカニがうろちょろ
GAME1

ペンペン迷路

こんなゲームに改造できるぞ！

マリオがのぼったりおりたり
GAME2

マリオ・ワールド

宇宙での一騎うち
GAME3

スターキラー

この本の内容についての問合せは、
往復ハガキか返信用封筒(60円切手添付)を同封して、
〒101 東京都千代田区神田錦町3-22
小笠原ビル4F

テクノポリス編集室

ファミリーベーシック入門係

☎03-295-4610

まで。なお、電話による問合せはできるだけ

月曜日～金曜日の午後5時～7時

の間にお願いします。

はじめに

ファミリーベーシックっておもしろいよ！

キミはもうファミリーベーシックを持っているのかな？　それともこれから買おうと思って、この本を見ているのかな？　どんなことがしたくてファミリーベーシックに興味を持ったんだろう。ファミコンのカセットみたいなゲームをつくりたい？　ベーシックを覚えて本格的なパソコンを使ってみたい？

ファミリーベーシックは、たぶんキミの期待にこたえられると思うよ。もちろんファミコンのカセットみたいにすごいゲームを作るのはむずかしい。でもカセットに入ってるのも結局だれかが作った「プログラム」なんだし、キミがこれから作ろうとしているのも「プログラム」だ。広い意味では同じものなんだよ。

ファミリーベーシックでおもしろいゲームを作るためには、この本に書いてあるようなことを知らなくちゃならない。でも“勉強するぞ！”なんてがんばらなくてもだいじょうぶ。ファミリーベーシックを使うのはとてもおもしろいことだから、楽しんでいるうちにキミはベーシックを覚えてしまうよ。カセットのゲームとはちょっと違った楽しみが味わえるってわけ。さあ、ファミリーベーシックしようぜ！

1 マリオを自由に呼び出せるぞ

P.27から

キャラクタ呼び出し術

2 いろんなマリオがぞーろぞろ

P.39から

どんなキャラクタも呼び出せる

3 P.59から 全自動でキャラを動かそう

プログラムしてみない？

P.77から

4 こんどはコントローラで!

キャラクタ自由自在操作術

5 敵が追いかけてくるよー

キャラクタの動かし方・つかまえ方

P.91から

6 かっこいい背景つくろうぜ

P.111から

BG-GRAPHICの使い方

7 いろいろやってみようよ！

P.135から

音楽・計算・エトセトラ

9 すぐに遊べるプログラム集

P.167から

オリジナル・ゲーム10本

8 V3のベーシックは強力だぞ!

P.151から

V3で拡張された命令と変更点

ファミリーベーシックのことば・さくいん

ベーシックの命令などの、意味や使い方を知りたいときは、右側に書いてあるページ数のところを見てください。ことばの読み方は上に書いてあるのを参考にしてね。また〔　〕の中は省略形です。省略形の使い方は90ページを見てください。

1

キャラクタ 呼び出し術!

ファミリーベーシックのなかには、楽しいキャラクタがいっぱいはいっているんだ。キミがキーボードからかんたんな命令を打ちこむだけで、すぐにブラウン管に出てきてくれるぞ。そんなキャラクタの呼び出し方や、きれいな色のつけ方をこの章でおぼえちゃおう。

マリオ、出てこい!

◆3つの命令でマリオ登場

ファミリーベーシックっていうのは、ファミコンにキミのやらせたいことを伝えるための言葉なんだ。これを使うとキミだけのオリジナルゲームを作ったり、そのほかいろんなことをファミコンにやらせられるってわけ。ベーシックはほとんどのパソコンで使っている言葉だから、ほかのパソコンを使うときの練習にもなるぞ。

ベーシックは、英語をもとに作られているけれど、単語の数が少ないから、すぐに覚えられるよ。このベーシックで、ファミコンにいろんなことを命令すると、ファミコンはキミのいうとおりにいろんなことをしてくれるし、キミの考えた新しいゲームだって作れるんだ。

まず、ためしにファミリーベーシックのカセットのなかにかくされているおなじみのキャラクタを呼び出してみよう。キャラクタを呼び出すには、3つの命令を使うんだ。

1番目はマリオを呼ぶ命令。

```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

0と0をまちがえたり、,と.をまちがえたりしないように注意して、このとおりに画面に出したらRETURNのキーを押そう。1字

あいているところにも注意していこう。

もし、?SN ERROR なんて出たら、これはどこかがまちがってるとコンピュータが教えてるんだ。もういちど正しく打ち直してから、RETURNキーを押そう。画面にOKと表示されたら成功だ。

次に、マリオの出現位置を決める命令だ。

```
SPRITE 0,120,120
```

と打ちこんで、同じようにRETURNキーを押そう。このように、命令を1つ打ちこむたびに必ずRETURNキーを押すこと！　OKと表示されたら成功。ここまでは、マリオのキャラクタを画面に出す準備だ。次の、3番目の命令でマリオが画面の真中あたりにポンと出てくるぞ。

```
SPRITE ON
```

そして、RETURNキーを押すと、OKと表示されてマリオが出てきたね。

次のページから、このベーシックをキミが使いこなせるようにくわしく説明していこう。今のところ、色がちょっと変だけど、色のつけ方も後で教えるよ。

マリオにつける番号札

◆アニメキャラクタは番号で指定する

DEF SPRITE0………は、
「キミを『スプライト0』と名づける!」
という命令だよ!

ステップ1で出てきた3つの命令の使い方を教えていこう。最初の命令は、

```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

だったね。この命令全体の説明は、ちょっと長くなるので、あとで説明するけれど、とにかく、ここで「マリオを〝スプライトの0番〟と名づける!」と命令しているんだと思ってほしい。この命令を打ちこんだら、マリオのこのキャラクタは、〝スプライトの0番〟という名前で扱われることになるのだ。

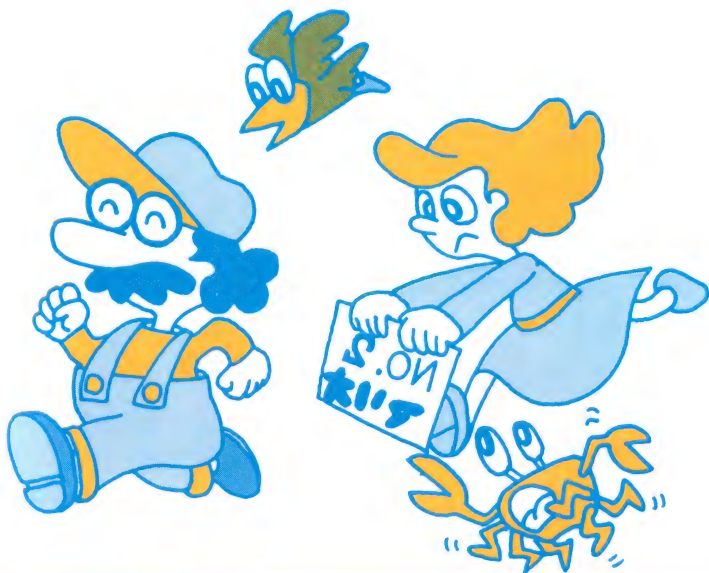
だから、2番目の命令にも、SPRITE 0……というふうに、出てくるね。この2つのSPRITEという言葉は、スプライト(アニメキャラクタ)の番号を指定するための目印だったんだ。

このように、スプライトにつけられた番号のことを、ふつう「ス

プライト番号」というよ。アニメキャラクタを呼び出して表示するには、必ずこのように番号をつけなくちゃいけないのだ。スプライト番号は、呼び出したアニメキャラクタにつける番号札のようなものだね。ベーシックは、番号札のついたキャラクタだけを表示してくれる仕組みなんだ。

スプライト番号は、スプライトの0番からスプライトの7番まで、8つ（0，1，2，3，4，5，6，7）ある。「たった8つ」なんてがっかりすることないよ。やってみればわかるけれど、これで十分なんだ。たりなくなったら、必要のない番号札をつけかえればいいだけ。

スプライトにつける番号には、もうひとつ「動作番号」というのがあるけれど、これはあとのお楽しみ。

ステップ3

マリオの引っ越し、自由自在

◆画面は256×240の方眼紙

最初の命令でマリオに〝スプライトの０番〟というスプライト番号をつけた。２番目の命令でどこに表示するかを決めるのだけれど、表示する位置を決めるのも数字なんだ。

画面にはなにも線が書いてないけれど、横が256マス、縦が240マスの方眼紙だと想像してみよう。そのマス目、ひとつひとつに〝座標〟が決められているんだ。横の位置を決めるのがＸ座標、縦の位置を決めるのがＹ座標。ただし算数や数学でならう座標とちがって、Ｙ座標の０は左上にあるので注意しよう。

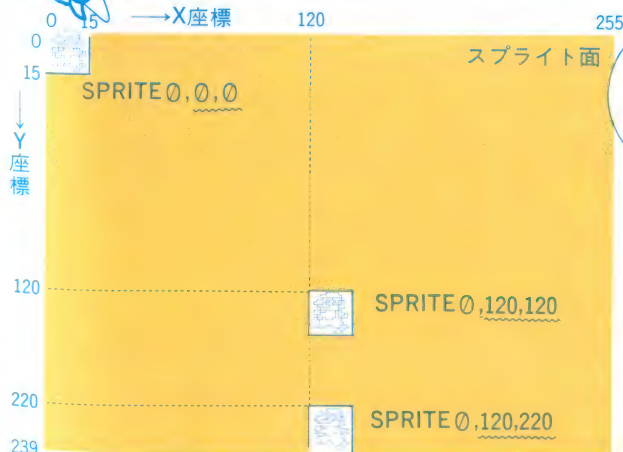
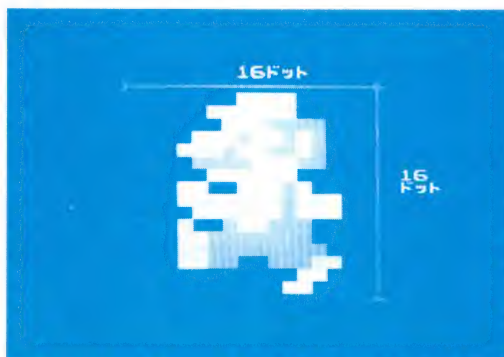
画面の〝方眼紙〟の１つのマス目のことを「ドット」という。だから、画面は256ドット×240ドットでできているともいうよ。

２番目の命令は、左から０，１，２;……と数えて120ドット目で、上から数えて120ドット目の座標（Ｘ座標が120、Ｙ座標が120）に、スプライト番号０のキャラクタを表示しなさいという意味なんだ。

SPRITE 0,	120,	120
スプライト番号0のキャラクタ(マリオ)を	左から数えて120ドット	上から数えて120ドットの座標に表示せよ

SPRITE ０のあとについている２つの数字はマリオ（スプライト番号０）の出現する座標だったというわけだ。だから、ここの数

字を変えればマリオを引っ越しさせられるよ。他の2つを同じように命令してあれば、SPRITE 0, 0, 0 RETURN（これでマリオは画面左上に）とか、SPRITE 0, 120, 220 RETURN（これでマリオは真中の下のほうに）などと命令するだけでいろんなところにマリオが現れるね。

マリオは16ドット×16ドットの大きさがあるので、この命令で指定する座標は、マリオの左上の1ドットの位置だということに注意しよう。だから、座標が、(0, 0)のときマリオの右下の部分は(15, 15)のところにひっかかっているよ。

スイッチ入れなきゃ見えないよ

◆SPRITE ONとSPRITE OFF

ここまでで説明した「スプライト」は、実は座標を決めてもまだ見えないようになっているんだ。前の2つの命令でマリオのキャラクタを呼び出して真中にかきこんでいるんだけれど、スプライトのスイッチがはいっていないので、座標を決める命令を打ちこんでもまだ見えなかったのだ。そこで、「スプライトのスイッチを入れろ」とファミコンに命令しなくちゃいけない。

それが、3番目のSPRITE　ONという命令なんだ。"ON"という言葉は、よくスイッチなんかについている"ON"と同じ言葉だ。

つまり、SPRITE ONとは、スプライト（のスイッチ）をオンにしろ！」ということなんだね。

これと反対の命令が、SPRITE OFFだ。"OFF"はもちろん、スイッチの"OFF"と同じ言葉。マリオがまだ画面に出ていたら、

SPRITE OFF

と打ちこんで、RETURNキーを押してごらん。マリオがパッと消えただろう。これはスプライトのスイッチを切ったのだ。スイッチを切っただけだから、マリオの絵そのものはまだ残っているよ。そこで、もういちど、SPRITE ON RETURNと命令してみよう。また、前と同じマリオが現れたね。この2つの命令でスプライトのスイッチを入れたり切ったりしてるということがわかったかな。

SPRITE ONでスイッチを入れると、次にSPRITE OFFでスイッチを切ったりしないかぎり、スプライトのスイッチは入ったままになっているよ。

ステップ1では最後にこの命令を入れたけど、実はいちばん最初に命令してもよかったんだ。ふつう、プログラムを作るときは最初にスイッチを入れておくことが多いみたいだね。

図1-4
SPRITE ONとSPRITE OFF

マリオの住む世界 ―スプライト面

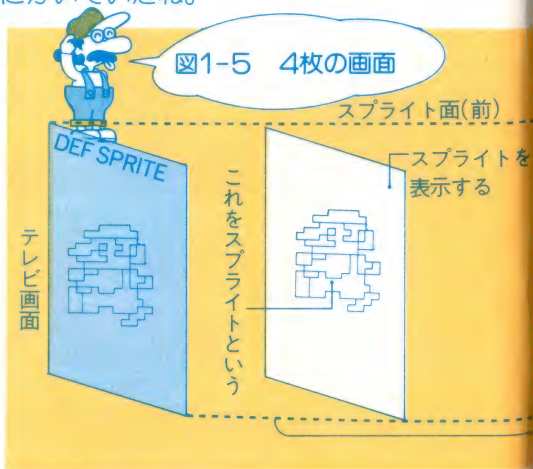
◆ファミコンが使う4枚の画面

3つの命令を打ちこんでいて、いちばんよく出てくる言葉は？それはSPRITE だね。

スプライトとは、英語で「妖精」という意味なんだ。妖精みたいに画面に現れたり、動きまわったりするキャラクタのことを、ベーシック用語で、スプライトというんだ。今はまだ動かし方を説明してないけれど、このスプライトはかんたんに動かせるんだよ。

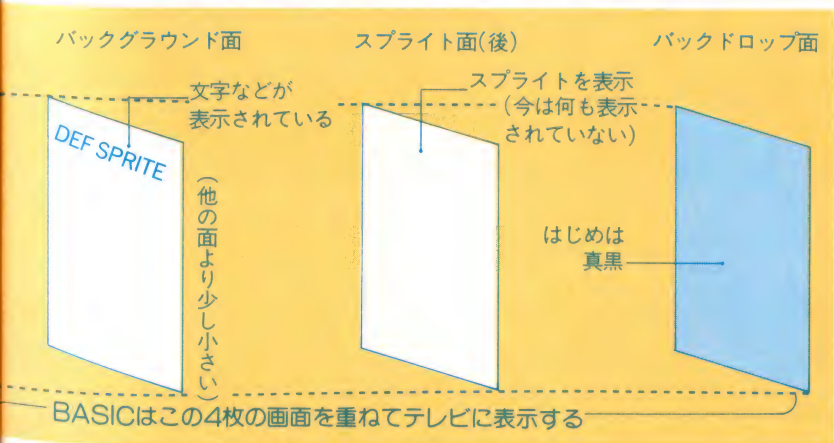
ところで、スプライトが表示される画面と、文字が表示される画面を前のページでは別々にかいていたね。

ちょっと不思議な話なんだけど、ファミリーベーシックをつないだファミコンの画面は、4枚の画面を重ねあわせたものなんだ！（これはファミリーベーシックの取扱説明書にも書いてあるよ）

右の図を見てね。

１つは、「バックドロップ面」という、いちばん奥にある面。これは、絵をかくときの画用紙みたいなもので、画面全体の背景だね。ベーシックをはじめる最初は、ただ真黒なだけだけど、これも命令で色をとりかえることができるんだ。

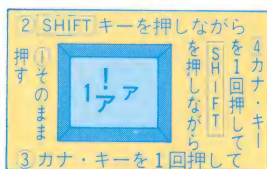
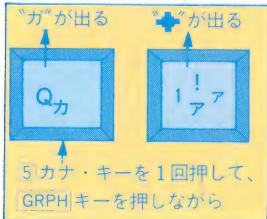
もう１つは、文字の出てくる「バックグランド面」。BG-GRAPHICで描いた背景の絵も、この面に表示されるんだ（背景の絵の描き方や表示の仕方は６章で教えるよ）。

そして、このバックグラウンド面の前と後ろに１枚ずつあるのが「スプライト面」。ここに、スプライト（マリオなどのアニメキャラクタ）が表示されるんだ。スプライトのスイッチをONにするってことは、このスプライト面のスイッチを入れて見えるようにすることだったんだね。

バックドロップ面とバックグラウンド面、そして２枚のスプライト面。あわせて４枚の画面を、ファミリーベーシックでは重ねあわせてテレビ画面に出しているというわけだ。

キーボードの使い方 その1

文字や記号の打ち込み方

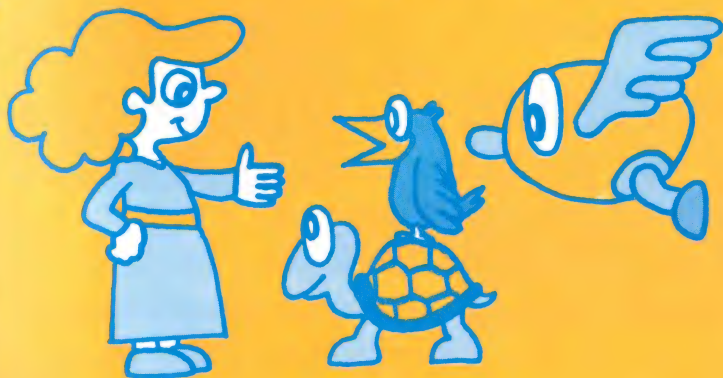
数字やアルファベットはそのまま押せば出てくるけど、記号やカナはちょっと工夫がいるね。ファミリーベーシックで遊ぶには、このキーボードの使い方がとてもたいせつなんだ。いろいろ試してみれば、とてもかんたんなことだよ。

①そのままキーを押す、② SHIFT キーを押しながらキーを押す、③カナ・キーを1回押してキーを押す(カナ・キーを2回押すともとの状態にもどるよ)、④カナ・キーを1回押したうえにSHIFTキーを押しながらキーを押す——この4つの方法で出てくる文字はそれぞれ、キーの左に書いてある文字、キーの上に書いてある文字、キーの下に書いてある文字、キーの右に書いてある文字——というきまりがあるのだ。上や右側に文字の書いてないものは②④の方法でキーを押しても出てこないということだ。これだけわかれば、キーボードに書いてある文字はみんな打ちこめる。

文字の出し方のきまり

もう1種類だけ、特別な方法で入れる文字がある。カタカナの濁音（ガやザ）は、⑤カナ・キーを1回押して、左下にある、GRPH キーを押しながら、カやサのキーを押して出すんだよ。

ついでにおなじやり方で、カナのいちばん上の列（ア〜ル）も押してごらん。いろんなケイや四角が出てくるよ。

いろんなマリオがぞーろぞろ

どんなキャラクタも呼び出せる

マリオが自由に呼び出せるのはいいけど、いろんな数字やカッコがついていて、ベーシックってめんどくさいなあ、なんて思ってる人いるかな？　この数字はいろんなマリオを登場させるためのものだから、めんどくさがっちゃダメ。いちど覚えちゃえば楽しいよ。

どんなマリオを呼ぼうかな

★スプライト番号の使い方

次のステップに入るまえに、画面の文字を掃除しよう。SHIFTキーを押しながら、CLR HOMEキーを押せば、カーソルがいちばん上にもどって、文字が消えるよ（スプライトは消えない）。さて、SPRITE ONでスプライト面のスイッチを入れたら（もう入れてある人は必要なし）次の命令を打ちこんでみよう。ステップ1でやったのをちょっと変えただけの命令だ。

```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
OK
SPRITE 0,10,150
```

OKが1つずつ出て、マリオが画面の左下に表示されたね。

```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
OK
SPRITE 0,10,150
OK
```

さて、今度はカーソル・キーを使って、リストの上のところまでカーソルをもどし次のように修正しよう。

```
DEF SPRITE 1,(0,1,0,0,0)=CHR
$(4)+CHR$(5)+CHR$(6)+CHR$(7)
```

スプライト番号とCHR$( )のなかの数字が5カ所変わっているだけだからかんたんだね。このとおりになったら、RETURNキーを

押そう。次に、またカーソルを動かして、リストを次のように修正しよう。

```
SPRITE 1,30,150
```

修正がおわったら RETURN 。すると、形のちょっとちがうマリオ（キャラクタテーブルAのWALK2）が、最初のキャラクタのとなりに出てきただろう！

これは、修正した命令でスプライト番号と呼び出すキャラクタを変え、そのスプライト番号のキャラクタを少し右側のほうに動かしているのだ。スプライト番号というものの便利さがわかったかな！

SPRITE 0 のキャラクタが残っているのは、スプライトはいちど命令したら、消す命令をしないかぎり画面に残っているからだ。

さあ、わかったら同じようにして、この命令の数字を少しずつずらして、マリオをいっぱい出してみよう。(CHR$( )のなかの数字はキャラクタテーブルAを見て変えてね)。

おっと、ただし、スプライトは横にならべて4つまでしか表示できない。5つ目は消えちゃうのだ。だから、5つ目からはY座標のところを170くらいにしておこう。

ところで、スプライトを1つだけ消す命令を教えよう。もし、スプライト番号0のキャラクタを消したかったら、SPRITE 0 RETURN と命令すればOK。0だけが画面から消えたね。3番を消したかったら、SPRITE 3 RETURN だ。

スイッチがいっぱい!!

★複雑な命令も、スイッチの集まり

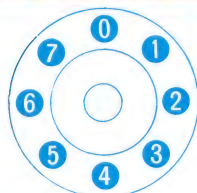
いよいよ、これまで出てきた命令のうちでいちばん長くて、いちばん大切なDEF SPRITE の命令を説明しよう。この命令は、大きくわけて3つの部分からできている。

DEF SPRITE0，までがステップ2で設明したように、アニメキャラクタにスプライト番号（ここではSPRITE 0）をつけている部分。これは、すぐわかるね。ついでにいうと、DEFという言葉は"定義する"という意味の英語（define）からきた言葉だ。

次の（0，1，0，0，0）は、スプライトの状態を決めるチャンネルの部分だ。カンマ(，）で区切られた数字は、ひとつひとつがチャンネルみたいなもので、ここをいろいろ変えると、スプライトの色や大きさが変わったり、キャラクタが裏返ったり、ひっくり返ったりする。ちゃんと覚えると、すごくおもしろく使える、大切

DEF SPRITE 0, (0,1,0,0,0) =CHR$(0)

0 1 2 3 4 5 6 7
スプライト番号

0 1 2 3
配色
合成型
オン
優先度
オフ
X反転
オフ
Y反転
オフ

左上
0
キャラクタ呼出し

な部分だよ。でもこれは少しややこしいからくわしいことは、次からのステップで説明しよう。

最後の＝CHR$(0)＋CHR$（1）＋CHR$（2）＋CHR$（3）が、マリオのキャラクタを呼び出している部分だ。ファミリーベーシックの取扱説明書に「キャラクタテーブルA」というのがあるね。その表に出ているキャラクタには、4隅に、番号がついているけど、この番号をCHR$(　)のなかに入れてキャラクタを呼び出しているんだ。“マリオ（WALK1）”と書かれたキャラクタには、0，1，2，3という4つの数字がついているね。この数字がこの部分に入っているんだよ。

この3つの部分がひとつの命令になって、ステップ1で画面に現れたような形のキャラクタ（マリオ）を、SPRITE　0と名づけているのだ。

ずいぶん、複雑そうだけど、よく説明を読めばかんたんだよ。キミが、テレビのスイッチを入れて、チャンネルをかえたり、ボリュームをかえたりするのと、おなじことなんだ。見た目は、英語の文章とか数学の式に似てるけど、これは、ただチャンネルやスイッチがならんでいるだけなんだね。次のステップから、このスイッチの使い方を説明していこう。

マリオの色を変えちゃえ!

★色の組合わせを変える「配色番号」

DEF SPRITE命令のカッコのなかの使い方をいよいよ教えよう。ステップ2のイラストを見ればわかるように、わりと単純なスイッチが集まっているんだ。この数字は左から順番にキチンと位置と意味が決まっているので、勝手にカッコのなかの数字を増やしたり、減らしたり、決められた数以外の数字を入れるとコンピュータは動いてくれないから注意しよう。でも、正しい方法で数字をいろいろと変えてみれば、キャラクタの楽しさがぐんと広がるぞ。このように、チャンネルやスイッチのような働きをする数字のことをコンピュータ用語でパラメータ(媒介変数)というよ。

では、カッコのなかに入っているパラメータを左から順に実験しながら説明していくぞ。

●DEF SPRITE　0, (0, 1, 0, 0, 0) ……

カッコのなかのいちばん最初の数字は、配色番号だ。使える数字は、0、1、2、3の4つ。試しに、

```
DEF SPRITE 0,(2,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

```
SPRITE 0,10,150
```

と命令してみよう。マリオが赤っぽくなったね。この部分を１や３に変えて試してごらん。少しずつ色が変わるね（SPRITE 0，10，150も必ず命令すること！）

ファミリーベーシックで使うキャラクタは３色でぬりわけるようになっているのだけど、その３色の組み合わせを決めるのがこの配色番号なんだ。つまり、３色の組み合わせには４通りあるわけだ。

でも、実はまだある。４通りの組み合わせを１セットとすると、スプライト用に３セット、背景用に２セット用意されているのだ。キャラクタテーブルの〝カラーチャート〞というところに、そのセットが全部出ているので見ておこう。

ただ、いちどに使える配色の組み合わせはやっぱり４通りしかない。組み合わせのセットを取りかえるには、あとで説明するCGSETという命令が必要なんだ。

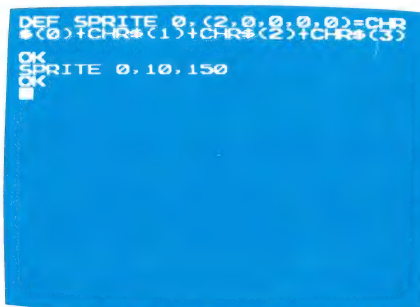
CGSETという命令を使っていないときの組み合わせは、カラーチャートのバックグラウンド用パレットコード１になってるよ。配色番号を０にしたときのマリオは、白、水色、青。２にしたときは白、オレンジ、赤の配色だったんだ。テレビによって色の出方がちがうこともあるけど、よく見比べてみよう。

マリオの大きさは変わる？

★大きさを決める「キャラクタ合成型」

●DEF SPRITE0，（0，1，0，0，0）………

スプライトには、実は2種類の大きさがある。マリオなんかは4文字分の大きさだけど、1文字分の小さいもの(レーザーがその例)もあるんだ。スプライト番号0のスプライトがそのどちらかを決めるのが、このキャラクタ合成型なのだ。0なら1文字分、1なら4文字分というわけだね。マリオは4文字分のキャラクタだから、ここでは1になっていたのだ。もし、ここが0だったら……？　さっそく試してみよう。命令の、カッコのなか2番目の数字（1）を0に修正してRETURN。そして、SPRITE 0，10，150の命令も、RETURN。写真と同じようになっていれば実験は成功だ。

```
DEF SPRITE 0,(2,0,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
OK
SPRITE 0,10,150
OK
```

1文字分のスプライトしか出ないね。それもよく見ると、マリオが小さくなっているんじゃなくて、マリオのキャラクタの左上4分の1だけが表示されている！　つまり、キャラクタテーブルAで0という番号のついた部分だけなんだ。

マリオは4文字分の大きさだといったけれど、もっと正確にいうと、1文字分のスプライトが4つ集まっているだけなんだ。命令の＝（イコール）のあとに、4つもCHR$(　)がついているのはそのせいなんだね。キャラクタ合成型を0（1文字分）にしたときは、このイコールのあとにCHR$(　)は、1つでよかったんだ。

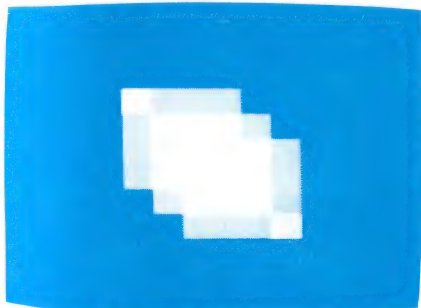
たとえば、レーザーを出したいときは、次の2つの命令になるよ。

```
DEF SPRITE 0,(0,0,0,0,0)=CHR
$(211)
```

この命令の次に、

```
SPRITE 0,10,150
```

この命令を打ちこめば、左下のほうにななめのレーザーが出てくるってわけだ。

▶レーザーの拡大

でもアニメキャラクタは、ほとんどが4文字分のキャラクタだから、このキャラクタ合成型もたいてい1にしておくことが多いね。

文字の後ろをくぐらせよう!

★スプライト面のどちらか選ぶ「表示優先度」

●DEF SPRITE0，（0，1，0，0，0）……

1章のステップ5でスプライトの表示されるスプライト面はバックグラウンド面（文字表示面）の前と後ろに1枚ずつあると説明したね。この表示優先度のパラメータは、0なら前のスプライト面、1なら後ろのスプライト面にスプライトを表示するものなんだ。

前のスプライト面にスプライトがあるときは文字よりも前に見えるし、後ろのスプライト面にあるときは文字の後ろに見えるよ。実験してみよう。

```
DEF SPRITE 0,(0,1,1,0,0)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

おなじみの命令だけど、カッコのなかの3番目の数字が1であることに注意してね。

```
SPRITE 0,16,24
```

ここでやった命令はスプライトを文字画面の左上すみに表示するために座標の値を（16，24）にしたんだけど、このために文字と、スプライトが重なったね。どっちが上にあるかな？　文字のほうが上にあるね！　これは、表示優先度のパラメータを1にしたので、

文字の面（バックグラウンド面）より後ろのスプライト面にキャラクタが表示されたからだ。では、次に表示優先度のパラメータを0に修正して命令してみよう（これはまえと同じ）。

RETURNキーを忘れずにね。

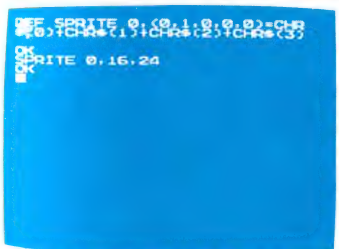
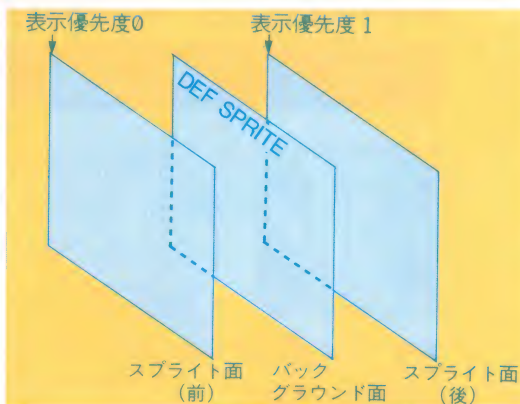
```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

```
DEF SPRITE 0,(0,1,1,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
OK
SPRITE 0,16,24
OK
```

▲1にしたら

```
DEF SPRITE 0,(0,1,0,0,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
OK
SPRITE 0,16,24
OK
```

▲0にしたら

画面に残ってるSPRITE 0,16,24のところにカーソルを移動させてRETURN。すると、同じ位置に同じキャラクタが現れたけど、今度はマリオの後ろに文字がかくれてしまった。前のスプライト面にマリオが移ったからなんだ。

今は、文字だけだけど、文字の面には背景画も表示されるので、このパラメータを0や1にすることで、スプライトが背景にかくれたり、背景の上に見えたりするよ。つまり、遠近感が出せる便利なパラメータというわけだ。

反対向きのマリオだよ!

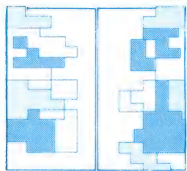
★キャラを左右逆にする「X軸方向反転指示」

●DEF SPRITE　Ø，（Ø，1、Ø，Ø，Ø）……

4番目のパラメータはちょっとおもしろいぞ。ここを1にするとキャラクタが左右逆になるのだ。ただし、このパラメータを変えるときには、＝（イコール）のあとにならんでいるCHR$( )の順番も変えないと変な表示になるよ。

というのも、マリオなどのアニメキャラクタは4つの部分でできているけど、このパラメータは、その1つひとつを左右にひっくりかえすからなんだ。

```
DEF SPRITE 0,(0,1,0,1,0)=CHR
$(0)+CHR$(1)+CHR$(2)+CHR$(3)
```

```
SPRITE 0,120,120
```

こんなマリオが出てくる。この場合は、左と右のキャラクタ部分を入れかえてやればいいんだ。

左と右を入れかえるには、CHR$( )の1番目と2番目を入れかえてやればいい（図参照）。つまり、

```
DEF SPRITE 0,(0,1,0,1,0)=CHR
$(1)+CHR$(0)+CHR$(3)+CHR$(2)
```

に修正(しゅうせい)して命令(めいれい)すればいいんだ。これで、SPRITE 0，120，120を命令(めいれい)すると、こんなふうに、ちゃんと右向(みぎむ)きのマリオが出(で)てくる仕組(しく)みになっているんだ。

CHR$(キャラクタダラ)( )のならんでいる順番(じゅんばん)も、こんなふうに大切(たいせつ)な意味(いみ)があるので、図(ず)を見(み)て、表示(ひょうじ)の位置(いち)とCHR$( )の順番(じゅんばん)を、よく覚(おぼ)えておこうね。

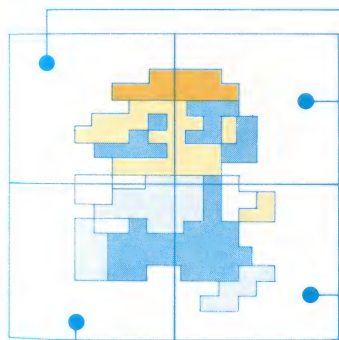
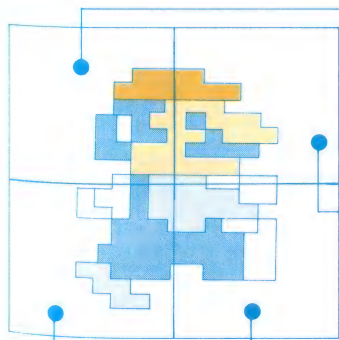



図2-6 CHR$( )の順番

DEF SPRITE 0,
(0, 1, 0, 0, 0)

ⓐ反転していないとき

=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
左上　右上　左下　右下

DEF SPRITE 0,
(0, 1, 0, 1, 0)

=CHR$(1)+CHR$(0)+CHR(3)+CHR(2)
左上　右上　左下　右下

ⓑ反転させたとき
(CHR$の順序を
変えている)

こんどは逆立ちマリオだ!

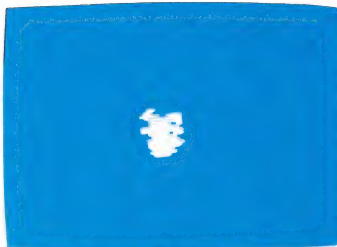
★キャラを上下逆にする「Y軸方向反転指示」

●DEF SPRITEØ，（Ø，1，Ø，Ø，Ø）……

これは、前の「X軸方向反転指示」と同じように、このパラメータを1にすると、キャラクタが上下逆になるんだ。ただし、やっぱり前と同じように、CHR$(　)のならべ方も変なくちゃいけないよ。図2-6をじっくり見てどうならべ変えればいいのか考えてみて。上と下とを入れかえればいいんだから、1番目と2番目のCHR$(　)を3番目と4番目のあとに持ってくればいいんだね。

```
DEF SPRITE 0,(0,1,0,0,1)=CHR
$(2)+CHR$(3)+CHR$(0)+CHR$(1)
OK
SPRITE 0,120,120
```

◀マリオの逆立ち

▶右向きもできるよ

と命令すれば、はい、マリオの逆立ち！

X軸方向反転指示とY軸方向反転指示の両方とも1にして、CHR$( )をうまくならべ変えれば、右向きの逆立ちもできるよ。

キーボードの使い方 その2

F1〜F8キー

キーボードのいちばん上にならんでいるF1〜F8のキーはファンクション・キーといって、1つで何文字分かのキー操作をいっぺんにやってくれる便利なキーだ。下のような働きがあるから、ときどき使ってみよう。この表はKEYLIST命令で見られるし、KEY命令で登録もできるから、取扱説明書を調べてやってみてね。

KEYLIST	
F1 LOAD(M)	テープに入ったプログラムの呼び出し
F2 PRINT	画面に文字や背景のキャラクタを表示する
F3 GOTO	行番号(3章で説明するよ)へ飛ぶ
F4 CHR$(	スプライト命令によく出てくる文字
F5 SPRITE	とちゅうで止めたプログラムを再開する
F6 CONT(M)	プログラムを画面に呼び出す
F7 LIST(M)	プログラムの実行を命令する
F8 RUN(M)	

マリオの色が本物になった!

★CGSET命令でパレットコードを指定

キャラクタの色は、配色番号0〜3の4通りの組み合わせで1セットになっているということはステップ3で説明したね。スプライト用には、3セット用意されていて、パレットコード0〜2という番号がついているんだ（取扱説明書のカラーチャート参照）。

ところが、そのままだといくら配色番号を変えてみてもバックグラウンド用のパレットコード1の配色しか使えない。マリオの色がいつまでたっても、キャラクタテーブルAのようにならなくて、つまんなかったんじゃないかな。

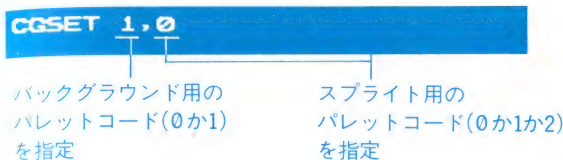
そこで、出てくるのがCGSETという命令だ。ステップ1の2つの命令をもういちど命令して青っぽいマリオを表示させておこう。そして、次の命令を打ちこんでくれ！

```
CGSET 1,0
```

どうだい？　マリオが本物の色になっただろう！　CGSETという命令は、パレットコード（配色番号の1セット）を指定する命令なんだ。

この命令も、2個のパラメータを使っているけど、実は最初のパラメータはここでは関係ないんだ。というのも、このパラメータは

バックグラウンド（背景や文字）のパレットコードを決めるものだからだ。

▼CGSET命令のパラメータ

マリオの色を本物にしたのは２番目のパラメータだ。このままの状態で、配色番号のパラメータ（カッコのなかの最初の０。ステップ３参照）を、０から１に変えて命令しなおすと、カラーチャートにあるようにマリオがルイージに変わるぞ。

同じように、スターキラーをこのままの色で出すには、パレットコードが１、配色番号が１だから、次のように命令すればいい。

```
CGSET 1,1
OK

DEF SPRITE 0,(1,1,0,0,0)=CHR
$(152)+CHR$(153)+CHR$(154)+C
HR$(155)
```

CGSETやDEF SPRITEの数字が変わっているのに気づいたよね。CHR$( )の中身もスターキラーを出すように変えてあるよ。キャラクタテーブルAのスターキラー(左の数字と見比べてみよう。

もっといろんな色使いたい!

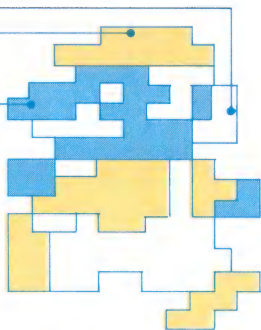
★好きな色を組み合わせられるPALETS

さて、もういちど画面の真中あたりに本物の色をしたマリオを出してほしい（ステップ8を見直せばできるよね)。

今度は、マリオの色や画面全体の色を自由自在に変える命令だ。CGSETと配色番号で指定できる色の組み合わせでも十分に遊べるけど、好きな色を選ぶことだってできるんだ。キミ好みの色が使えるってわけだ。画面にマリオが出てるかな？　そこで、次の命令、

```
PALETS 0,22,15,33,48
```

を命令してみて。まちがいなく命令できたら、テレビ画面が真赤になって、マリオが黒人のおじいさんみたいになったね。

この命令は、今使っているパレットコードの配色を自分の好きな色に変えちゃう命令だ。最初のパラメータで指定された配色番号（0～3）の色の合わせを、3番目から5番目のパラメータで順に指定しているんだ。

色を指定する数字(色コード)は、取扱説明書に出ている。上のほうが青い色、下のほうが赤い色、左が暗く、右が明るい。実際にはどんな色かパラメータを変えて確かめてみるのもいいね。

それには、2番目のパラメータ(バックドロップ面の色)を変えるのがわかりやすい。これは画面全体の色を指定しているのだ。カーソルを動かして2番目のパラメータだけ変えてみよう。たとえば、

```
PALETS 0,17,15,33,48
```

と変えて、RETURNキーを押すとバックが青に変わったね。

ここを0から60まで変えてやってみれば、色と色コードの関係がだいたいわかるはずだ。もし、やってるとちゅうで文字の色が見えなくなったら、CTRLキーを押しながらDキーを押せば、もとにもどるよ。ただし、スプライト面も消えるのでもういちどSPRITE ONしよう。

最後に、もとの色にもどしてみようか。

```
PALETS 0,15,54,22,2
```

で、もとにもどったね。

キーボードの使い方――その3

特殊なキー

RETURN ESC STOP CTRL CLR HOME INS DEL

RETURN　このキーは、命令や行の区切りで押すたいせつなキーだ。これを押しわすれると、命令が通じないよ。

ESC　長いリストが画面に出てきたとき、このキーを押すといったん止まるよ。なにかキーを押せばまた、ずらずらっと出てきはじめる。プログラムを見直すときに便利なLIST命令の一時停止なんだ。

STOP　プログラムの実行を止めるキー。

CTRL　このキーを押しながら、アルファベットのキーを押すとおもしろい働きをするよ。CTRLとDを押すと、スプライトの状態をはじめにもどす機能。CTRLとEならカーソルから右にある文字を消してしまう。CTRLとGなら"ピッ" という音。

```
10 SPRITE 0,(0,1,0
10      SPRITE 0,(0,1,
10 DEF SPRITE 0,(0,1,
```

```
10 SPRITEE, 0,(0,1,
10 SPRITEE 0,(0,1,0
10 SPRITE 0,(0,1,0
```

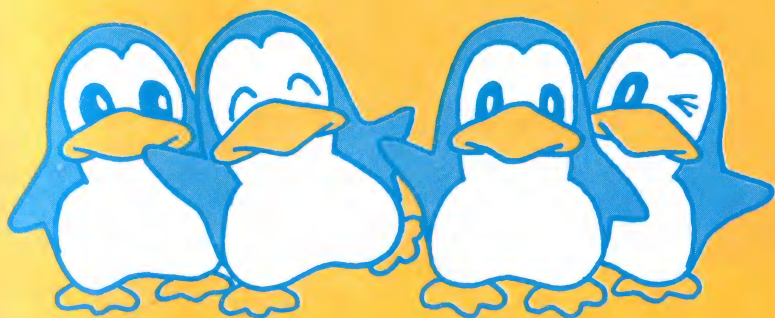
CLR HOME　このキーだけを押すと、カーソルがホームポジション（左上）にもどるだけだけど、SHIFTキーを押しながらCLR HOMEキーを押すと画面の文字や背景が全部消えちゃうよ。

INS　このキーを押すたびにカーソルが文字ごと右に移動して、左側に空白ができるよ。書きわすれた文字を入れていくんだ。

DEL　このキーを押すたびにカーソルが文字ごと左に移動して、左側の文字を消していくよ。いらない文字を消すために使うんだよ。

プログラムしてみない？

全自動でキャラを動かそう！

キャラクタの呼び出し方、バッチリ、わかっちゃったかな？　今度はいよいよ本格的に、キャラクタを動かす方法を覚えちゃおう。それには、「プログラム」の作り方をマスターすればいいんだ。全自動でキャラクタが動くよ。

プログラムでファミコンへ指令

♠プログラムとRUN命令

今までやってきた命令は、1回1回、ファミコンに命令していくやり方だね。でも、これでは、せいぜいキャラクタを出すことしかできない。

ゲームを作るためには、これとはちょっとちがったやり方で「プログラム」を作るんだ。これは、たくさんの命令をいっぺんにファミコンにわたすための方法なんだよ。9章にはいっている楽しいゲーム・プログラムは、こうして作ったのだ。またファミコンのほかのゲームも、ベーシックじゃないけど、やっぱりプログラムでつくってあるんだよ。

でも、プログラムといっても、そんなに変わったことをするわけじゃない。ただ、1つ1つの命令の頭に数字をつけていくだけだよ。

第1章のいちばんはじめで覚えた、3つの命令をプログラムにしてみよう（順番をちょっと入れかえているけど意味は同じ）。

リスト3-1

```
1 SPRITE ON
2 DEF SPRITE 0,(0,1,0,0,0)=C
HR$(0)+CHR$(1)+CHR$(2)+CHR$(
3)
3 SPRITE 0,120,120
```

もういちど、1章のステップ1でやったのと見比べると、ただ番号がついて順番が変わっただけだね。それに、3つとも打ちこみおわっても、ファミコンが動いてくれないのもちがうところだ。

このプログラムというものは、ファミコンへの指令書、命令を書いた手紙みたいなものなんだ。

この命令どおりに働いてもらうには、

```
RUN
```

RETURNと命令すればいい。やってごらん。まえと同じことをしてくれただろう！　このRUNという命令は「プログラムどおりにしろ！」という命令なんだ。

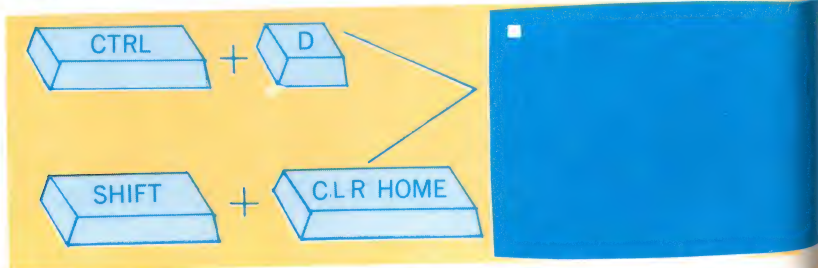
プログラムはこんなに便利!

♠行番号とLIST命令

リスト3-1の頭についている数字は「行番号」といって、プログラムにとって、とても大事なものなんだ。コンピュータは、この行番号の順番に命令を実行してくれることになるからだ。それに、こうしてプログラムの形で命令を伝えておくと、たとえ画面から命令が消えてしまっても、いつでも何回でもRUN RETURNで、命令を実行してもらえるんだよ。

リスト3-1のプログラムでちゃんとマリオが出てきたら、今度はCTRLキーを押しながらDキーを押そう。スプライトが消えたね。次に、前にもやったようにSHIFTキーを押しながら、CLR HOMEキーを押そう。これで画面には左上のカーソル以外なにもなくなったね。そこで、またRUN RETURNと命令してみよう。また命令をきいてくれて、マリオを出したね。では次に、

LIST

```
LIST
1 SPRITE ON
2 DEF SPRITE 0,(0,1,0,0,0)=C
HR$(0)+CHR$(1)+CHR$(2)+CHR$(
3)
3 SPRITE 0,120,120
OK
```

と命令してみよう。ほら、さっきの命令（プログラム）がまた出てきたね。このLISTという命令は､｢プログラムを出しなさい｣という命令なんだ。こうして画面に出てきたプログラムや本にのっているプログラムのことも､｢リスト｣というよ。これもよく使う言葉だから覚えておこう。そしてこの本でもこれから出てくるリストには番号をつけていくことにするよ。

じゃあ手はじめにリスト3-1のプログラムをちょっと修正してみようかな。プログラムの修正もリストを出してかんたんにできるよ。

```
2 DEF SPRITE 0,(0,1,0,0,0)=C
HR$(96)+CHR$(97)+CHR$(98)+CH
R$(99)
```

リスト3-2

これが修正のお手本。

INS キー（わからなかったら58ページを見よう）を利用して、こんなふうに修正し、RETURN キーを押せば修正OK。行ごとに必ずRETURNキーを押すのを忘れずにね。

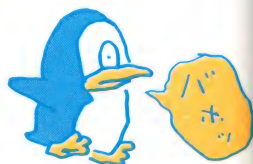
これでRUN RETURN とすれば、今度はペンペンが出てくるぞ。もういちど、LIST RETURN として、プログラムリストを見てごらん。ちゃんと直っているだろう？

まちがえちゃった！でも平気さ

♠行番号は10番とびにつけていく

プログラムは命令を順番にならべたものだってことがわかりやすいように、行番号をはじめ1，2，3とつけたよね。でも9章のプログラムを見ると10，20……が多い。他のプログラムもだいたいそうなってる。ふつう、プログラムを作りはじめるときは、行番号は10，20とつけていくんだ。「えっ、じゃあ、1番とか2番はどうなるの」と不思議かもしれないね。コンピュータは、1番とか2番が抜けていても、ちっとも気にしないで、とにかく行番号の小さなほうから実行していってくれるのだ。

それに、10，20とつけていくといいことがある。それは、いいわすれた命令をあとからつけくわえやすいってこと。

ペンペンを出すプログラムを次のように変えてみよう。①まず、行番号のところにカーソルを移動して、1を10に変えて[RETURN]、2を20に変えて[RETURN]、3を30に変えて[RETURN]。そこでLIST [RETURN]でリストを見てみよう。行番号1，2，3，10，20，30の命令が出てきたね。1，2，3は前の命令がまだ残っているんだ。

②そこで、1 RETURN、2 RETURN、3 RETURNと打ちこんで、もういちどリストを出そう（LIST RETURNだね）。今度は、行番号10，20，30だけになった。

```
1 SPRITE ON
2 DEF SPRITE 0,(0,1,0,0,0)=C
HR$(96)+CHR$(97)+CHR$(98)+CH
R$(99)
3 SPRITE 0,120,120
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,0,0)=
CHR$(96)+CHR$(97)+CHR$(98)+C
HR$(99)
30 SPRITE 0,120,120
OK
```

こんなふうにかんたんに行番号をつけかえることができる。そして、いらない行番号は、ただその数字をRETURNで入れていくだけで、消えちゃうんだ。

この行番号なら、行番号5に新しい命令をあとから入れても行番号10の命令より先に実行してくれるし、行番号15なら行番号10と20のあいだで実行してくれるのだ。もちろん、5や15じゃなくても、1，2や16，17でも同じことだよ。INSやDELのキーを使えば、命令もかんたんに直せるし、こんなふうに行番号をとばしておけばつ加えるのもかんたん。消すのもすぐできるしね。プログラムっていうのはこんなにかんたんに直せるんだよ。

きれいな画面でペンペンが走る!

♠CLS命令とFOR〜NEXT命令

```
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,0,0)=
CHR$(96)+CHR$(97)+CHR$(98)+C
HR$(99)
30 SPRITE 0,120,120
```

リスト3-4A

このプログラムをRUNさせたときに、画面に残っている文字がきれいに掃除されるようにしてみよう。

```
5 CLS
```

と命令を加えて、RUNさせてみて。リストがパッと消えたね。CLSは文字(背景画も)を消す命令なんだ。でも、プログラムは消えないから安心してね。リストを出してみると、ちゃんと行番号の小さい順に出てくるぞ。

さてこのへんで必殺技その1。FOR〜NEXT命令の使い方を教えよう。これはいろんな形で使える便利な命令なんだ。まず、その使い方の1例として、ペンペンを左から右にサーッと走らせてみよう。いままでに打ちこんだリストを、次のように変えればいい。①行番号20の命令を右向きのペンペンに変える(X軸方向反転のパラメータを1に変え、CHR$( )をならべかえる。50ページを見てね。

②行番号30の命令のX座標の部分をアルファベットのXにおきかえる。③そして、次の2つの命令を加える。

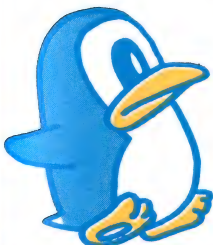
```
25 FOR X=0  TO 240
40 NEXT
```

そこで、リストを出してみよう（LIST RETURN）。次のようなリストが出てきたかな。

リスト3-4B

```
5 CLS
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,1,0)=
CHR$(97)+CHR$(96)+CHR$(99)+C
HR$(98)
25 FOR X=0  TO 240
30 SPRITE 0,X,120
40 NEXT
```

これをRUNさせれば、ペンペンの走りが見られるのだ。

▼リスト3-4Bのプログラムを RUNすると……

全自動で数が増えていく

♠変数とFOR～NEXTの使い方

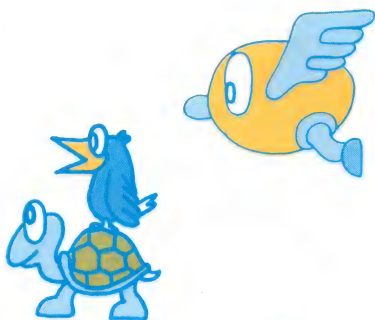
リスト3-4Bの25行、30行、40行の使い方を教えよう。

30行の命令は、スプライト番号0のキャラクタ（ペンペン）を表示するものだということはわかっているよね。でも、前は数字が入ってたところにあるXってなんだろう?

これは、Xという名前の黒板のようなもので、アルファベットだけど中身がいろんな数字に変わるものなんだ。命令次第で自動的に数字が書きこまれ、それまで書いてあった数字は消されてしまうんだと考えてほしい。そ

図5-5
変数は命令次第でいろんな数になる

してこのXなどの名前を「変数名」という。変数名は別にXじゃなくても、A～Zのどれを使ってもいいし、2文字でもいいんだよ。

そして、この変数Xの値（数の大きさ）を決めているのが25行と40行の命令なんだ。

25行は、「Xを0から240まで（1ずつ）増やせ」という命令。まず、0からはじめて30行の命令を実行し、40行のNEXT（これは英語の「次」）でXを1増やして（つまり0から1にして）25行にもどり、また30行の命令を実行する。そして、また40行でXを1増やして（つまり2にして）、25行にもどり、30行の命令を実行……この繰り返しをXが240を超えるまでやっているんだ。

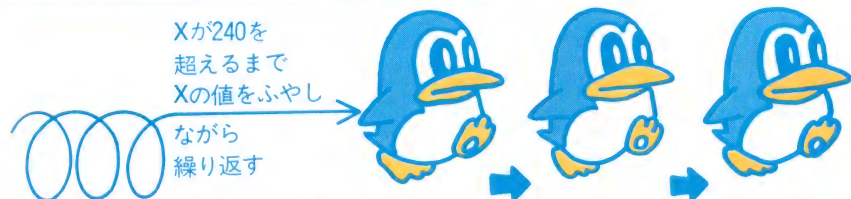
ここでちょっと実験。25行にSTEP 4と加えて、

25 FOR X=0 TO 240 STEP 4

にしてみよう。そして、RUNすると、今度はペンペンが速くなったね。これは、Xの増え方を1ずつではなく4ずつにしたためだ。STEPのあとにつけた数の分ずつ増えるんだよ。

```
25 FOR X=0 TO 240 STEP 4
```

リスト3-5

Xが240を
超えるまで
Xの値をふやし
ながら
繰り返す

スプライト面のX座標の値(X)を変えながら次々にペンペンを表示していく

ペンペン、ちょっと速すぎるよ!

PAUSE命令でひとやすみ

FOR……のあとに、STEPを加えるとペンペンが速くなるのはわかったね。では、今度はペンペンをおそくしてみよう。まず、リスト3-5の修正をもとにもどして、リスト3-4Bと同じにしておいてね。

そこで、31行に新しい命令をつけくわえてみよう。

```
31 PAUSE 5
```

リスト3-6

このとおりにうちこんで、RETURNキーを押せばいいよ。リストを出すと（LIST RETURN）、ちゃんと30行と40行のあいだにはいっているかな?

これでRUNさせてみると、ペンペンがゆっくりになったね。

PAUSEという命令は、ちょっと命令の実行を待ってもらう命令なんだ。よくカセットデッキについているPAUSEというスイッチと同じ意味だね。

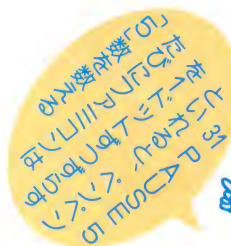
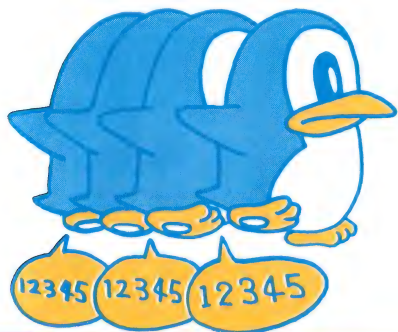
PAUSEのうしろについている数字は、

どのくらい待つかを決める数字だ。数字を大きくすると、長いあいだ待つようになるし、小さくするとちょっとしか待たないよ。この数だけ、ファミコンが数を数えて待ってるんだね。でも、ファミコンはすごく速く数を数えるので、5といってもほんのわずかの時間

だ。だって、リスト3-6のプログラムは、1ドット、ペンペンを動かす（つまりペンペンを16分の1だけ右にずらす）たびに5数えてるんだよ。

PAUSEのあとについている数字をいろいろ変えて、RUNさせてみれば使い方がよくわかるよ。

それから、もうひとつ。PAUSEのあとになにも数字をつけないで、31 PAUSEとしてみよう。これでRUNさせると、ペンペンは動かない。でも、なにかキーを押すたびに少しずつ（1ドットずつ）動いていくよ。数字のないPAUSE命令は、「なにかキーを押すまで待ってて！」という命令になるんだ。

STEPとこのPAUSEをうまく組みあわせれば、ペンペンがいろんなスピードで動いていくよ。

本格アニメにしちゃおう!

スプライト番号を活用して動かす

ペンペンが走るといっても、同じ格好でただすべってるだけじゃつまんないね。キャラクタテーブルを見ると、ペンペンの歩く姿には2種類あるから、この2つともを使ってアニメっぽくしてみよう。

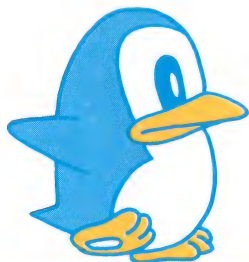
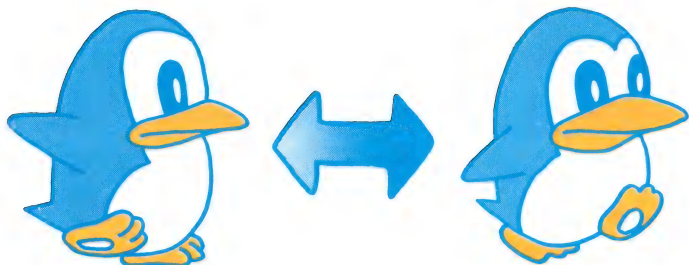
ここで、スプライト番号を活用するんだ。

新しく、スプライト番号1として、ペンペンの"左歩2"というキャラクタを右向きにしたものを呼び出してみよう。これは、DEF SPRITE……でやるんだったね。

行番号21として、次のような命令をリストにつけくわえよう。

```
21 DEF SPRITE 1,(0,1,0,1,0)=
CHR$(101)+CHR$(100)+CHR$(103
)+CHR$(102)
```

これで、スプライト番号1に、ペンペンのもうひとつの歩く姿が入ったぞ。スプライト番号0とスプライト番号1を交互に表示させれば、ペンペンが歩くように見えるはずだ。でも、片方を表示するまえに、もう片方を消すようにしないと2つが重なって見えにくくなるから注意。

上のようなことを考えて、リストを改造すると次のようになるじっくり見比べてね。

リスト3-7

```
5 CLS
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,1,0)=
CHR$(97)+CHR$(96)+CHR$(99)+C
HR$(98)
21 DEF SPRITE 1,(0,1,0,1,0)=
CHR$(101)+CHR$(100)+CHR$(103
)+CHR$(102)
25 FOR X=0  TO 240
30 SPRITE 0,X,120
31 PAUSE 10
32 SPRITE 0
35 SPRITE 1,X,120
36 PAUSE 10
37 SPRITE 1
40 NEXT
```

5 画面をきれいにする
10 スプライト面のスイッチオン！
20 スプライト番号0にペンペンのキャラクタ(左歩1の右向き)をセット
21 スプライト番号1にペンペンのキャラクタ(左歩2の右向き)をセット
25 変数Xを0から240まで1ずつ増やしてくりかえす
30 スプライト番号0のキャラクタを表示
31 「10」数えるあいだ休み
32 スプライト番号0を消す
35 スプライト番号1のキャラクタを表示
36 「10」数えるあいだ休み
37 スプライト番号1のキャラクタを消す
40 Xを1増やして25行にもどる

もっとダイナミックに動かそう!

♠プログラムの改造

今までのいろんな命令を応用すれば、リスト3-7をもとにしてペンペンの動きをいろいろ変えられるぞ。30行と35行のSPRITE…という命令が、うしろの2つの数でX座標とY座標を指定しているんだということがわかっていれば、あとはキミのアイデア次第だ。たとえば、30行と35行を次のように変えればペンペンがたてに動くね。

```
30 SPRITE 0;120;X
35 SPRITE 1;120;X
```

リスト3-8A

変数Xが今度はY座標のかわりになって、0から240まで増えるからなんだ。

また、25行、30行、35行を次のようにすると、ペルペンがはねているように見えるぞ。

```
25 FOR X=0  TO 240 STEP 16
30 SPRITE 0;X;120
35 SPRITE 1;X+8;136
```

リスト3-8B

ほかにも、たとえば行番号30、35を

```
30 SPRITE 0;X;X
35 SPRITE 1;X;X
```

リスト3-8C

とすれば、ペンペンはななめに動くね。

あと、いろいろ、キミのアイデアで試してみよう。

▲リスト3−8Aの修正をするとペンペンはたてに動く

▲リスト3−8Cの修正をするとペンペンはななめに動く

▲リスト3-8Bの修正をするとペンペンはとびはねる

文字変数と数値変数

変数には2種類あるよ

変数の見かけはいつもアルファベットか、アルファベットと数字・記号の組みあわせだ。そして、コンピュータが命令を実行していくにしたがって、この変数の中身も変わっていく。変数は、座標やスプライト番号などに使われているので、変数の中身が変わるということは座標やスプライト番号が変わるということ。つまり、画面のキャラクタなどが動くということになるんだね。

こんなふうに、中身が数字の変数のことを数値変数というよ。

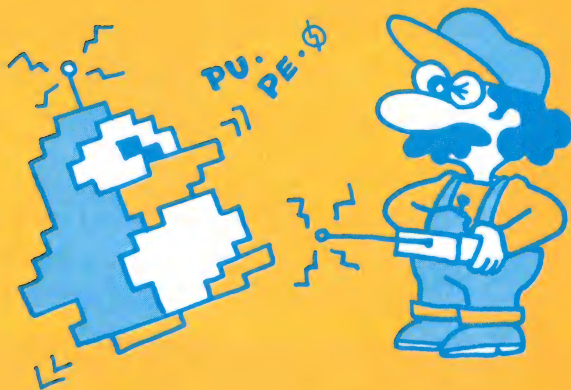
これとは別に、文字変数というのがある。文字変数は、あとで出てくる音楽を演奏するPLAY文や、文字の表示などのときに出てくるものだ。ほかにも、いろんなデータをうまく処理するために使われるので、数値変数と同じようにたいせつ。

数値変数と文字変数は、使われ方は似ているけど、まったくちがうものなので、決して同じような扱い方はしてはいけないんだ。この2つをたしたり、引いたり、等号でつないだりすると、エラーが出るよ。だから、この2つはとても見わけやすいようになっている。A，B，FX，FY，X1，Y1などアルファベットと数字だけのものが数値変数で、文字変数には必ず最後にA$，FX$，X1$などと$マークがついているんだよ。

キャラクタ自由自在操作術

こんどはコントローラで動かそう

今まではキャラクタが勝手に動いていたけど、今度はコントローラを使って自由自在に動かす方法を教えちゃおう。いよいよ、プログラムも本格的になってくるから、どんどんおもしろいことができるようになるぞ！

ステップ 1

コントローラがプログラムに参加!

STICK,STRIG,PRINT,GOTO

コントローラは、十字形のボタンやトリガーを押すたびに、信号をファミコンに送ってるんだ。その信号を数値に変えて、プログラムのなかに呼びこんでしまうのが、STICK( )とSTRIG( )という「関数」。関数とは、決められたルールを守りながら数値や文字を作り出す変数の一種だよ。

ややこしい説明より、かんたんな例をひとつあげよう。

今までのプログラムを、NEW RETURNと命令して全部消してしまおう。それから、次のプログラムを打ちこんでみよう。

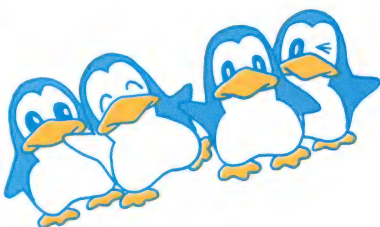
```
10 PRINT STICK(0),STRIG(0)
20 GOTO 10
```

リスト4-1

このPRINTという命令は、そのあとにかかれた変数に入っている数を画面に出す命令だ。変数をカンマ(,)やセミコロン(;)で区切ってならべると、いくつもの変数をいっぺんに表示できるよ。リスト4-1では、カンマを使ってみた。こうすると、2つの変数の値をはなして表示してくれるんだ。

20行のGOTOという命令は、この命令のあとに行番号を書くとその行番号から実行してくれる。リスト4-1では、まず10行を実行して次に20行を実行すると、また10行にもどって……と、永遠に

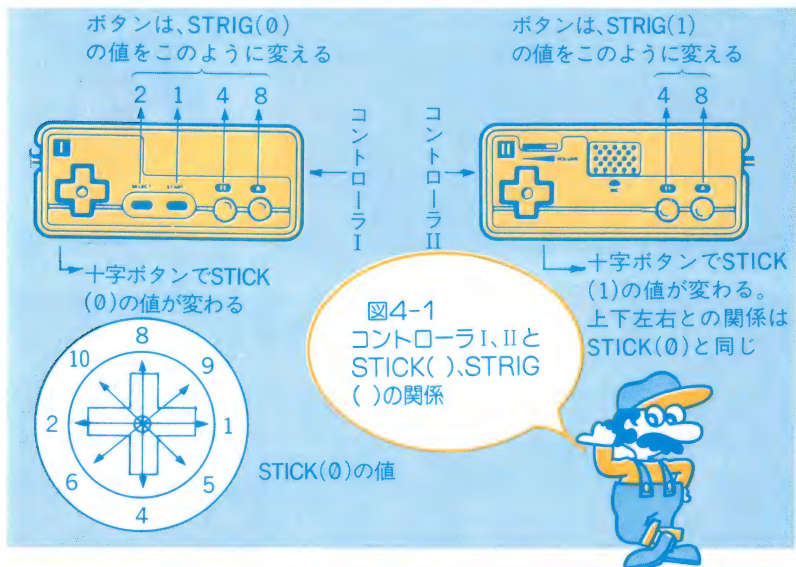
プログラムを繰り返すようにしてあるんだ。

このプログラムをRUNさせてごらん。0が2本の筋になって出てきたね。この左側が、STICK（0）の値、右側がSTRIG（0）の値なんだ。コントローラⅠの十字ボタンを押せば、図4-1にかいてあるとおりに数字が変わるね。

さあ、わかったら、プログラムを止めよう。右上の STOP というキーを押せば、止まるよ。カッコのなかの2つの0を1に変えれば、今度はコントローラⅡに反応するぞ。

こんなのおもしろくないと思う人がいるかもしれない。でも、これを使うと、いよいよコントローラでキャラが動かせるんだよ。

ボタンは、STRIG(0)
の値をこのように変える
2 1 4 8
コントローラⅠ
十字ボタンでSTICK
(0)の値が変わる
8
10 9
2 1
6 5
4
STICK(0)の値

ボタンは、STRIG(1)
の値をこのように変える
4 8
コントローラⅡ
十字ボタンでSTICK
(1)の値が変わる。
上下左右との関係は
STICK(0)と同じ

図4-1
コントローラⅠ、Ⅱと
STICK( )、STRIG
( )の関係

コントローラでアキレスを動かす

IF文とプログラムのテクニック

STICK(0)を使って、アキレスを動かしてみよう。今までのプログラムをNEWしてから、次のプログラムを打ちこんでね。

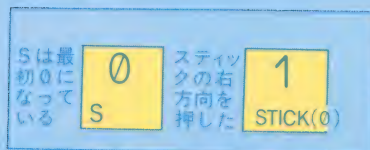
リスト4-2

```
10 SPRITE ON:CLS:CGSET 1,1
20 DEF SPRITE 0,(0,1,0,0,0)=
CHR$(64)+CHR$(65)+CHR$(66)+C
HR$(67)
30 S=STICK(0)
40 IF S=0 THEN SPRITE 0,120,
120
50 IF S=8 THEN SPRITE 0,120,
30
60 IF S=1 THEN SPRITE 0,210,
120
70 IF S=4 THEN SPRITE 0,120,
210
80 IF S=2 THEN SPRITE 0,30,1
20
90 GOTO 30
```

RUNさせると、コントローラＩの十字ボタンを押せば、アキレスの位置が変わるようになる。

10行は知っている命令ばかりがならんでいるね。：（コロン）は1つの行番号のなかにいくつかの命令を入れたいときに使うんだ。

20行は、もちろん、アキレス（左1）の呼び出し。

30行で、STICK(0) の値をSにコピーしているんだ。つまり、

変数Sの値を、STICK(0)の値と同じにする命令。このやり方は、これからもよく使うので覚えておこう。

40行～80行で使っているIF　○○○　THEN　×××は、「もし、○○○なら、そのとき×××せよ」という命令。よく見ていくとわかると思うけど、たとえば、40行は「もしSが0なら（つまり十字ボタンを押してないということだね)、そのときスプライト0を座標（120,120）に表示せよ」という意味だ。もしちがう場合は、すぐ次の行に移ることになっている。

そして、90行でまた30行にもどり、Sの値を入れなおしてから、下の行を実行して、また何度でも繰り返しているんだ。

ステップ3

アキレスが一瞬はばたいた

変数の値を取りかえるSWAP命令

いよいよ、アキレスをコントローラで自由自在に動かすプログラム作りに入ろう。今までのプログラムをNEWで消して、まず、次のプログラムを打ちこんでね。

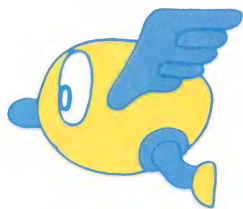
リスト4-3A

```
10 SPRITE ON:CLS:CGSET 1,2
20 X=120:Y=70:IMA=0:MAE=1:P1
=0:P2=1
30 DEF SPRITE 0,(2,1,0,0,0)=
CHR$(64)+CHR$(65)+CHR$(66)+C
HR$(67)
40 DEF SPRITE 1,(2,1,0,0,0)=
CHR$(68)+CHR$(69)+CHR$(70)+C
HR$(71)
100 SPRITE MAE:SPRITE IMA,X,
Y:PAUSE 5
400 MAE=IMA:IMA=P2
420 GOTO 100
```

10行のCGSET命令や30,40行の配色番号を工夫してアキレスの色を本物にしてるのには気づいたかな？

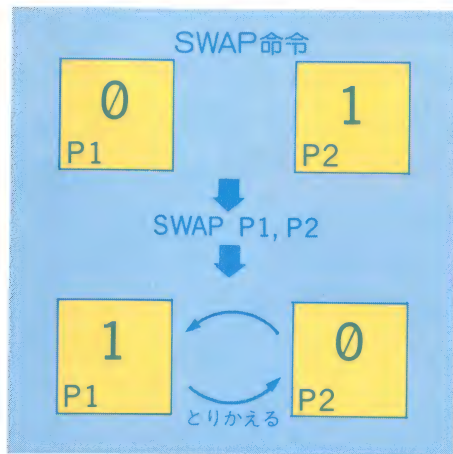
20行は、このプログラムで使う変数の最初の値を決めているよ。X，Yはスプライトの座標。IMAやMAEはスプライト番号に使う変数。P1，P2はスプライト番号を変えるために使うんだ。

10行、20行のように、画面の状態や変数の最初の値（初期値）を決める部分を「初期設定」というから覚えておいてね。

100行では、変数を使ってスプライト消去とスプライト表示をやってるね。これは72ページでやった。そして、400行で変数の値を変えている。IMAに入っている数をMAEに移して、かわりにIMAの値をP2にしてるね。そして、420行で100行の命令にもどる。RUNさせてみると、一瞬だけど、アキレスが1回はばたいて見えるのは、400行で変数の値を変えているからなんだ。

行番号がとびとびになっているのは、あとからもっと命令をつけくわえようと思っているから。

さて、STOPキーでプログラムを止めて、リスト4-3Aに次の行を加えよう。

```
410 SWAP P1,P2
```

リスト4-3B

この命令は、2つの変数の値を取りかえるよ。この命令を実行するたびにP1とP2は値を取りかえる。さて、これでRUNすると……？

ステップ4

アキレスは飛んでゆく

X座標を変えてアキレスを動かす

SWAP命令を加えただけでリスト4-3Aのプログラムがアキレスのはばたきのプログラムになったね。こんなふうに、変数の値を変える命令を加えていくことで、アキレスを動かすことができるんだ。今度は、リスト4-3A、BにSTICK（0）を参加させて、コントローラで操作できるようにしてみよう。

次の命令を新しくつけくわえよう。

```
200 S=STICK(0)
210 IF S=2 THEN X=X-4
```

これを入れてRUNさせると、コントローラIの十字ボタンを左に押せば、アキレスがはばたきながら左に飛んでいくぞ。

Sが2のとき（つまりスティックの左を押したとき）、X座標の値を4減らしているからこうなるんだ。この命令のあと、420行まで行って、100行にもどり新しいX座標のところにアキレスを表示してるわけだ。画面で、キャラクタが動くのはこういう仕組みなんだ。

でも、やってみると、アキレスが画面の左端にきたところで、

?IL ERROR IN 100

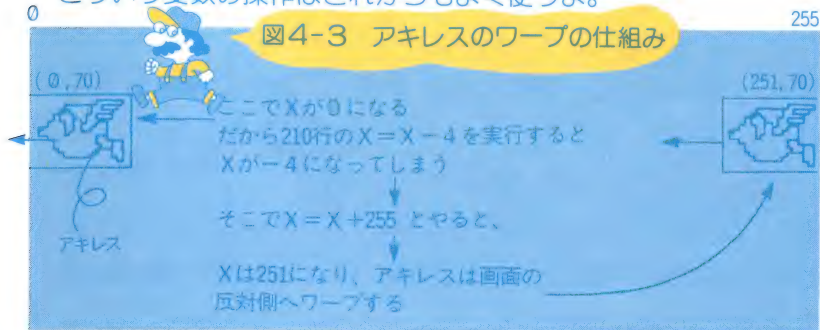
となるね。そこで、PRINT Xと命令してみよう。Xの値を表示させるんだ。出てきた数字は、－4！　エラーはこのせいだ。

X座標は0から255までしかないので、それ以外の数字がSPRITE命令のX座標に入ると、このエラーが出るようになっているのだ。これを防ぐには、Xの値がマイナスになったときに、Xの値を変えてしまえばいいんだ。

```
300 IF X<0 THEN X=X+255
```

もうわかったと思うけど、この命令で、「Xが0より小さくなったら（Xがマイナスになったら）Xに255を加えて、新しいXの値としろ」ということをファミコンに伝えているんだね。255を加えると、ちょうど画面の反対側にワープするぞ。

こういう変数の操作はこれからもよく使うよ。

図4-3　アキレスのワープの仕組み

舞いあがったり舞いおりたり

XとYを同時に変えればななめ

X座標の変え方とエラーの防ぎ方がわかれば、たて方向の動きはかんたんだね。今までのプログラムに次の命令を加えよう。

```
220 IF S=8 THEN Y=Y-4
230 IF S=4 THEN Y=Y+4
```

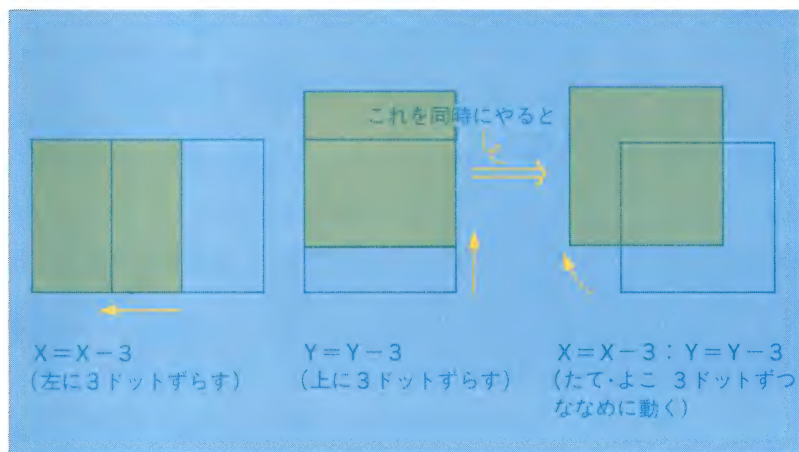
これで、上下に十字ボタンを押せばその方向に舞いあがるね。でも、また画面の上とか下でエラーがでるので、次の命令も入れよう。

```
320 IF Y<0 THEN Y=Y+239
330 IF Y>239 THEN Y=Y-239
```

これは、まえと同じように、Y座標が0～239の数以外になることを防いでいるのだ。310行の命令がまだないけれど、これは次のステップで完成させたときに入れるよ。

さらに、今度はななめにも動かそう。ななめ左上はSTICK(0)が10、ななめ左下は6だったね。だからななめに動かすには、次のプログラムを入れればいい。

```
240 IF S=10 THEN X=X-3:Y=Y-3
250 IF S=6 THEN X=X-3:Y=Y+3
```

この2つのIF文のTHEN以下にある命令で、X座標とY座標をいっぺんに変えているんだ。こうすると、ななめに動くよね。

XやYから引いたりたしたりする数を4にしてもまちがいじゃないけれど、ほかと同じように4にしてしまうと、ななめに動くときがたてやよこよりもちょっと速くなってしまうんだ。たて・よこ3ドットずつだとちょうどいいくらいなんだ。これは「ピタゴラスの定理」という有名な数学の定理で決めたんだ。学校の先生に「ピタゴラスの定理ってなあに」ってきけば教えてくれると思うよ。

ステップ 6

上下左右に飛びまわる

キャラクタを逆に動かすには

さて、こうなるとアキレスをコントローラで右に動かすことも同じようにしてできちゃうね。次の行を追加しよう。

```
260 IF S=1 THEN X=X+4
270 IF S=9 THEN X=X+3:Y=Y-3
280 IF S=5 THEN X=X+3:Y=Y+3
310 IF X>255 THEN X=X-255
```

260～280行がコントローラＩの十字ボタンからの信号を判断しているところで、310行はもちろん、エラーを防ぐための命令だね。

でも、これだとたしかに右にも動くけど、後ろ向きに飛んでいることになるね。そこで、右へ動くときは右向きのキャラクタが出るようにしよう。まず、右向きのキャラクタを定義する命令を加える。これは、もうわかるよね。

```
50 DEF SPRITE 2,(2,1,0,1,0)=
CHR$(65)+CHR$(64)+CHR$(67)+C
HR$(66)
60 DEF SPRITE 3,(2,1,0,1,0)=
CHR$(69)+CHR$(68)+CHR$(71)+C
HR$(70)
```

次は、どうやって、変数IMAに、スプライト番号2や3を入れるかだ。これはちょっとテクニックが必要だ。新しく変数Pを導入しよう。この変数Pを、右方向なら2、左方向なら0になるようにしておいて、IMA=P2+Pとすればいいのだ。この変更はかんたん。今までのプログラムを次のように変更しよう。

210行と240行と250行のそれぞれの最後に：P＝0を加える。たとえば210行なら、こんなふうになる。

```
210 IF S=2 THEN X=X-4:P=0
```

次に、同じようにして、260行、270行、280行に：P＝2を加える。そして、400行を次のように変えるのだ。

```
400 MAE=IMA:IMA=P2+P
```

この変更がおわったら、アキレスは自由自在に飛びまわるぞ。バックを空の感じにしたい人は、次の行も加えてみよう。

```
15 PALETB 0,18,44,21,7
```

これでバックが空になっちゃうよ。

リスト4-3Aから今までの追加分を全部打ち込むと、アキレスが画面の中を自由に飛びまわるようになるよ。

▼ラククラク、プログラム法

命令は省略して使える

DEF SPRITE……とかCHR$(192)……とか、ときどき全部打つのがめんどくさくなってしまう人がいるかもしれないね。"SPRITE" や "CHR$(" はファンクション・キーを利用すればいいとして、他の命令などをラクに打ちこむ方法はないかな。

ちゃんとあるんだよ。ベーシックの命令などにはたいてい、"省略形" というものがあるんだよ。よく使われるものをちょっと紹介してみよう。

LIST──→L.　でOK。

PRINT──→？（またはP.）でOK。

GOTO──→G.　でOK。

FOR～TO～STEP…NEXT──→F.～TO～ST.…N.でOK。

LOCATE──→LOC.　でOK。

CGSET──→CG.　でOK。

まだまだほかにもいろいろあるよ。PRINT以外は、その命令の頭文字やいくつかの文字を取って、ピリオドをつけた形が多いね。これらはみんなベーシックの文法編に書いてあるから参考にしてね。

プログラムを省略形で打ちこんでも、次にリストを取ったときには、ちゃんと正式な形で出てくるから不思議。試してごらん。

キャラクタの動かし方・つかまえ方

敵が追いかけてくるよー

ここでは敵がキミの動かすキャラクタを追いかけてきたり、キミが敵をやっつけたりするプログラムを作っちゃう。SPRITE命令と似たMOVE命令という、便利な命令の使い方も教えるよ！

マリオとレディのラブシーン◆◆

♥キャラクタを動かすMOVE命令

ファミリーベーシックでは、SPRITE命令に似たもうひとつのキャラクタ用命令がある。それはMOVE命令だ。

これを使えば、すごくかんたんにキャラクタが動いちゃうのだ。ためしてみよう。1章のステップ1のときのように、ダイレクトモードで次の3つの命令を打ちこんでみて。

```
SPRITE ON

DEF MOVE(0)=SPRITE (1,3,1,12
7,0,2)

MOVE 0
```

最後の命令をくだしたとたん、レディが右に走り出したね。今までは、プログラムを組んで動かしていたのに！　3章で、ペンペンを歩かせたときの苦労に比べるとずいぶんラクチンだ。

実は、ここで動いているキャラクタもやっぱりスプライトなんだ。ただ、SPRITE命令のときのスプライトとは、また別種類のスプライトだと考えたほうがいいね。

MOVE（0）の0は、「動作番号」といって、スプライト番号な

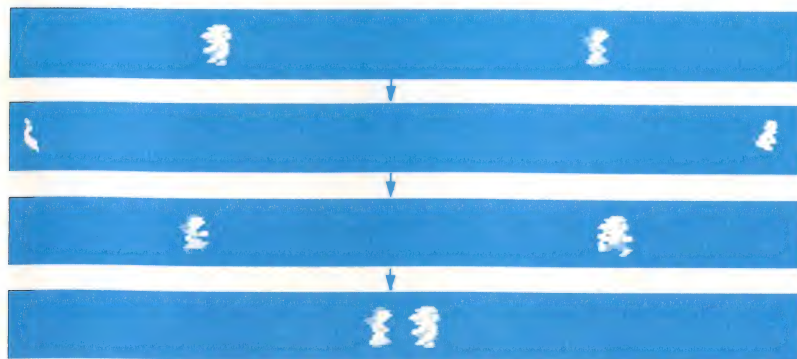
どと同じように、0～7まで使える呼び出し用の番号だ。そして、＝SPRITE……でどのキャラクタを、どんなふうに動かすかを決めているのだ（くわしくは次のステップで説明するからね）。

最後のMOVE　0は、動作番号0のキャラクタをスタートさせる命令なんだ。

さっそく、かんたんなプログラムを作ってみよう。

リスト5-1

```
10 SPRITE ON
20 DEF MOVE(0)=SPRITE (1,3,1
,127,0,2)
30 DEF MOVE(1)=SPRITE (0,7,1
,127,0,0)
40 MOVE 0,1
```

40行の命令は、動作番号0と1のキャラクタを同時に動かす命令だ。リスト5-1をRUNさせると、レディとマリオが同時に反対方向へ走り出して、最後にピタッと抱きあっちゃうよ。

キャラクタの切りかえスイッチ

♡DEF MOVEのパラメータ

ステップ1のDEF MOVE……は、MOVE命令のスプライトを決めているんだ。SPRITE……のあとに続く数字（パラメータ）が大切だ。これも、2章で説明したDEF SPRITEに続く数字のように、スイッチのようなものなんだ。

リスト5-1の30行のうち1文字だけ考えてRUNさせてみよう。カッコのなかのいちばん左の数字を2に変えるだけ。

```
30 DEF MOVE(1)=SPRITE (2,7,1
,127,0,0)
```

このように直ればOK。RUNさせると、今度は、マリオのかわりにファイターフライが出てきたぞ。つまりここは、キャラクタを選ぶスイッチなんだね。0～15のどれか1つの数字をこの場所に入れておくと、数字にあわせていろんなキャラクタが動くのだ。たとえば、0ならマリオ、1ならレディ、2ならファイターフライ……というように。今みたいに、数字を入れ直していろいろ出してみて。

他の数字（パラメータ）も，図のようにいろんな意味があるよ。リスト5-1をもとにして、他のパラメータも変えて試してみよう。ただし、図で書いてある数字の範囲以外の数を入れると、エラーになるから、注意！

図5-2DEF MOVEのスイッチ

```
DEF MOVE(0)=SPRITE(1, 3,1,127,0, 2)
```

動作番号0～7
スプライト番号と同じように、スプライトを呼びだすとき区別するためにつける番号

キャラクタの種類　0～15
この数字を変えると動くキャラクタが変わる

0→マリオ	8→スターキラー
1→レディ	9→スターシップ
2→ファイターフライ	10→爆発
3→アキレス	11→ニタニタ
4→ペンペン	12→レーザー
5→ファイアーボール	13→カメさん
6→車	14→カニさん
7→スピナー	15→トリさん

動作の方向指定　0～8
この数字を変えると動いていく方向が変わる。写真のように、0は静止、1は上、2はななめ右……と数字と方向が対応している

表示優先度　0～1
ここはDEF SPRITEの「表示優先度」(➡2章ステップ5)と同じ。ここが0だとバックグラウンド(文字や背景) の前にあるスプライト面、1だと後ろのスプライト面に表示される

動かす速さ　1～255
スプライトの動くスピードのレベル。1がいちばん速く、255はものすごくおそい

全移動量　1～255
1回のMOVE命令で動く距離。この数字が1増えるごとに、スプライトは2ドットずつ遠くまで移動する。たとえば255にすると、255×2=510ドット、つまりスプライト面全体の2倍の距離を動く

配色番号　0～3
これも、DEF SPRITEの「配色番号」と同じ。CGSETで指定されたパレットコードから配色を選ぶ(➡2章ステップ8)

動け! 止まれ! 消えろ!

♥MOVE命令とCUT、ERA

DEF MOVEで決めた動作番号のキャラクタに、動作スタートの命令をくだすのが、MOVE命令だ。「MOVE」のあとに動作番号を指定すると、その番号のキャラクタが動きはじめるのだ。

リスト5-1をRUNさせて、キャラクタが止まったところで、

と命令してみよう。動作番号0のキャラクタ、つまりここではレディが動きはじめる。MOVE 1と命令すれば、マリオが動くのだ。2つとも動かすには、リスト5-1の40行にあるように、カンマで区切って数字を続ければいいだけ。

MOVE命令で動きはじめたキャラクタは、DEF MOVEのSPRITE( )のなかにある4番目のパラメータ（全移動量）で決められた距離だけ動いてしまうまで止まらない。これを途中で止める命令が、CUT命令だ。リスト5-1をRUNさせて、キャラクタがまだ動いているうちに、

```
CUT 1
```

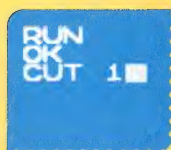
と命令してみよう。図のようにすればいいよ。

図5-3A動いているキャラクタをとちゅうでストップさせよう

```
RUN
OK
■
```

まずRUNさせると、カーソルがOKの下に出る。

そこでCUT 1と打ちこむ。

```
RUN
OK
CUT 1
```

カーソルをRUNのところにもどして[RETURN]。RUNの命令が入り、キャラクタが動きはじめる。

カーソルはすぐにCへ移るので、もういちど[RETURN]キーを押す。

CUT命令が入り動作番号1(マリオ)が止まる。

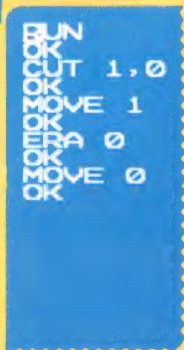
ここで、CUT　0，1とすれば両方止まるよ。動きを再開させるには、もういちど、MOVE　0　と命令すればいい。止まった位置から動きはじめて、「全移動量」の残りを動くのだ。

つぎにERA 0と命令してみよう。こうすると今度は、動作番号0のキャラクタ（レディ）が消えてしまう。ERA 1ならマリオが消えるね。両方いっぺんに消すには、ERA 0，1というふうに、入れればいいんだ。

図5-3B　MOVE、CUT、ERA、の実験

```
RUN
OK
CUT 1,0
OK
MOVE 1
OK
ERA 0
OK
MOVE 0
OK
```

リターン・キーを押すたびにカーソルが移動して命令を実行

ステップ4

ファイアーボールがあちこち動くぞ

♥変数の計算式の使い方

ある1つのキャラクタをいろんな方向に動かしたいときは、動かしたい方向の分だけ、DEF MOVEすればいいんだけど、これはプログラムの工夫でかんたんにできるよ。NEWしてから、次のリストを打ちこんでみてね。

これをRUNさせてから、SPRITE ONして、MOVE 0，1，2，3と命令すれば4つのファイアボールが2ドット分だけ動くよ。

リスト5-4

```
100 FOR I=0 TO 3
110 DEF MOVE(I)=SPRITE (5,I*
2+2,1,1,0,0)
120 NEXT
```

何回かMOVE 0，1，2，と命令していけば少しずつ動いていくのだ。110行で「全移動量」を1にしているからだね。ところで、

「動作方向」のところにあるI＊2+2ははじめて出てたきけど、これは変数を使った計算式なんだ。「＊」とは、計算や数学で使う「×」(かける)のこと。計算記号については7章を見てね。

ここでは、Iが1増えるごとに方向のパラメータが、2，4，6，8となっていくようにしてあるのだ。これはそれぞれ、右上、右下、左下、左上の指定になっているというわけ。それから、「全移動量」を1にしておいたのは、少しずつ動かしていたほうがあとで他のリストを組みあわせるときに便利だからだ。

でも、MOVEの使い方はこれだけじゃないんだよ。プログラムでの命令の使い方はいろいろ考えられるのだ。

図5-3 動作番号とファイアーボール

動作番号3(方向8)

動作番号0(方向2)

動作番号1(方向4)

動作番号2(方向6)

＊それぞれの方向に2ドットずつ進む

アキレスのスタートを決めるよ

♥RND関数とPOSITION命令

さて、次にファイアーボールが進んでいく目標を決めよう。ここでは、アキレスをいろんな位置からスタートさせて画面を2周させよう。

```
10 CLS:SPRITE ON:CGSET 1,2
20 DEF MOVE(4)=SPRITE (3,7,1
,255,0,2)
30 SX=RND(220)+16:SY=RND(200
)+24
40 POSITION 4,SX,SY
50 MOVE 4
```

リスト5-5

上のリストをリスト5-4に加えてね。

図5-5 サイコロみたいな
A　RND関数

RND(6)　＋　1

この数より少ない乱数を作る。つまり、0から5までの数のどれかがでたらめに出てくる

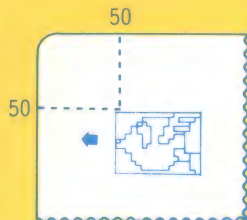
その乱数に1を加えると、1から6までのどれかがでたらめに出る

20行の意味はわかると思うけど、30行のRND( )ははじめてだね。これはカッコのなかの数より少ない乱数を作る関数なんだ。乱数というのは「でたらめの数」という意味で、もしRND(6)+1なら、サイコロと同じく、1から6までの数がでたらめに作れるんだ。

30行では、ほぼバックグラウンド面のなかのどこかにアキレスが出現するようにしているんだ。

40行の命令も、とても大切な命令だ。SPRITE命令では、座標を指定したね。MOVE命令でもスタートの位置をこの命令で指定できるんだ。

もちろんそのあとにMOVE命令が必要だよ。

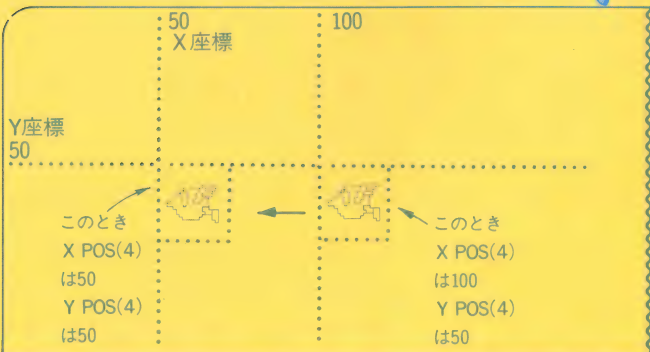
図5-5 B POSITION命令でスタート位置を決める

リスト5-5だけでもアキレスを適当な位置からスタートさせるプログラムになっているよ。まちがいなく打ちこめたら、RUNさせてみよう。

ステップ6

ファイアーボールが追っかけてくるよ!

♥XPOS、YPOSの使い方

さて、リスト5-1で決めたファイアーボールをアキレスのほうへ追いかけさせるにはどうしたらいいと思う? まず、飛んでいるアキレスの位置を知りたいね。そのための関数がXPOS( )とYPOS( )だ。カッコのなかに知りたいキャラクタの動作番号を入れると、この関数の値はそれぞれ、その動作番号がそのときいた座標の値に変わるんだ。そこで、とりあえず変数X, Yにアキレスが今いる位置の座標を入れておこう。そのための命令は次のようになる。

図5-6 関数XPOS( ), YPOS( )はアキレスの位置を教えてくれる

```
200 X=XPOS(4):Y=YPOS(4)
```

リスト5-6A

そして、またXPOS( )を使ってファイアーボールの位置を調べ、X，Yと比べてファイアーボールの動きを決めればいいんだ。FX,FYをファイアーボールの今の座標にして次のようなプログラムを今までのプログラム（リスト5-4，5-5，5-6Aをあわせたもの）につけくわえよう。ファイアーボールの動作番号は方向に応じて0～3に決めていた（リスト5-4）ので、それをFという変数にそのつど入れることにしたよ。

アキレスの座標がわかれば、ファイアーボールをどの方向に動かせばいいのかがわかるよね。その方向に動く動作番号を変数Fに入れてやれば、ファイアーボールがアキレスを追っかけていくのだ。

```
210 FX=XPOS(F):FY=YPOS(F)
220 IF FX<X AND FY>Y THEN F=
230 IF FX<X AND FY<Y THEN F=
240 IF FX>X AND FY<Y THEN F=
250 IF FX>X AND FY>Y THEN F=
300 POSITION F,FX,FY:MOVE F
310 IF MOVE(F)=-1 THEN 310
320 GOTO 200
```

リスト5-6B

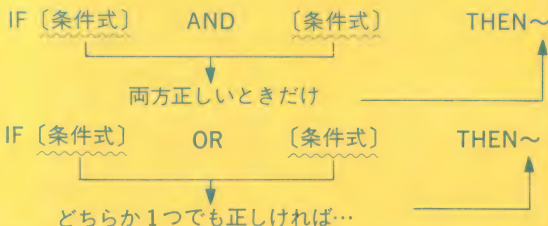
ファミコンが考えてくれるの？

♥ANDとORってなんだろう

ステップ6のプログラム、正直いって見ただけじゃよくわからなかったんじゃないかな。少し説明をしておこう。

220行から250行の命令は、よく使う大事なやり方だ。ここでは、ANDという言葉がはじめて出てきたね。

ANDとは、IF文のなかで使う言葉で、「そして」という意味だ。220行をふつうの言葉にすると「もし、FX（ファイアーボールのX座標）がX（アキレスのX座標）より小さく、FYがYより大きければ、F（ファイアーボールの動作番号）を0にしろ」と、なる。

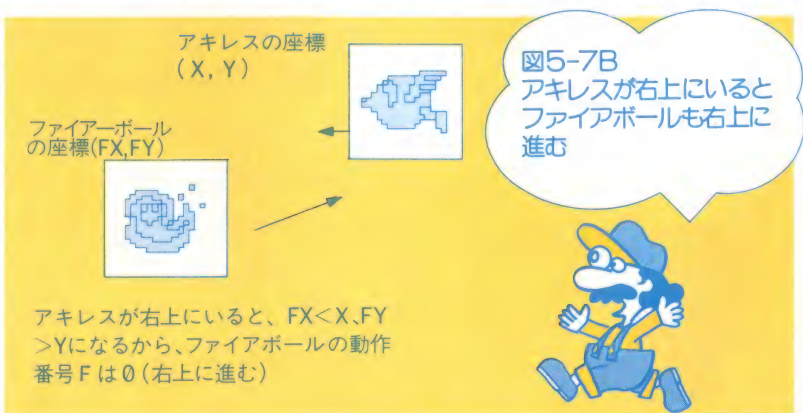
この場合、FX<XやFY>Yになっていても、FYがYより小さければTHEN以下の命令は実行されないのだ。つまり、ANDをは

さむ2つの条件式（FX＜XやFY＞Yのこと）が両方とも正しいときだけTHEN以下の命令を実行する仕組みになっているんだね。似たような言葉にORというのもあるよ（図5-7A）。

この220行の命令の仕組みは図5-7Bを見よう。これと同じことを220〜250行でやっているんだ。

300行のPOSITION命令は、動作番号が変わったときに、その場所から新しい動作番号のファイアーボールを動かすためだ。これがないと、動作番号ごとにバラバラに動いてしまうよ。

310行はファイアーボールが動きおわるのを待っているところ。動作番号Fが動いているあいだは、MOVE（F）という関数は、−1になっているんだ。止まったら、0になる仕組み。この命令がないと、前の動作番号が動きおわらないうちに次の動作番号が動きはじめるのでヘンになっちゃうんだ。THENのあとはGOTO　310と同じ意味。THENのあとでは、GOTOを打たなくてもいいんだ。

そして、320行でまた200行にもどって同じことを繰り返す。変数X、YやFX、FYやFが次々に変わって、画面を動かしてるんだ。

アキレスが右上にいると、FX<X、FY>Yになるから、ファイアボールの動作番号Fは0（右上に進む）

追いつかれちゃう、逃げろ! 逃げろ!

♥ERA命令の使い方とルーチン

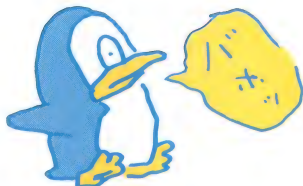
さて、リスト5-4~6BをあわせたプログラムをRUNしてみよう。あれ? ファイアーボールがときどきだぶって残ってしまうね。

これは、別々の動作番号で動かしているのが原因。前に動いていた動作番号は、次に300行の命令で呼び出されるまでその場所に残ってしまっているからだ。

だから、1つの動作番号を動かしおわったら前のは消してしまおう。今までのリストに、次の2行分を加えればだいじょうぶ。

```
205 FMAE=F
260 IF FMAE<>F THEN ERA FMAE
```

リスト5-8

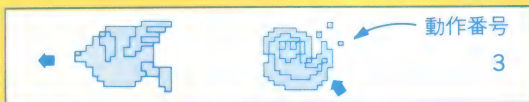
FMAE(ファイアーボールの前の動作番号)という変数を新しく加えて、これに今動きおわったばかりの動作番号を入れておくんだ。Fのほうは、220行からの命令で位置に応じて変えられるからね。260行の「<>」という記号は、左と右の変数や式の値が等しくないという意味だ。つまり、FMAE(前

の動作番号）とF（今度の動作番号）がちがっていたら、FMAEのほうを消してしまうということになる（図5-8）。

リスト5-8で一応完成だ。

ところで、200行から320行のように、まとまって1つのこと（ファイアーボールがアキレスを追いかける）をする命令のまとまりを「ルーチン」というんだ。さしずめ、ここのルーチンは、追いかけルーチンとでも名づけられそうだね。

さあ、RUNしてみよう！　ファイアーボールがアキレスを追いかけていくよ。やめるときは、STOPキーを押せばOK。

図5-8 前の動作番号を消す

300行のMOVE F(Fは3)でこうなる

次にFMAEはF、つまり3になり、
Fは240行で2に変わる。
FとFMAEの値はちがうので、
ERA FMAE(FMAEは3)を実行。

そして、300行でMOVE F(Fは2)。

これを繰り返す。

アキレスがつかまった

♥つかまったかどうかはABS関数で

リスト5-4から5-8で作ったプログラムは、つかまってもそのままだったね。じゃあ、つかまるとメッセージを出しておわるようにしてみよう。

つかまったかどうかの判定に便利なのがABS( )という関数だ。これは、カッコのなかに入った数式や変数の値がマイナスでもプラスにかえてしまう関数だ。

```
PRINT ABS(-10)
```

と命令してごらん。すぐ下に10と出てきたね。

PRINT ABS(3−10)と命令すると、7と出てくるよ。カッコの式の答えがマイナスになってもプラスにしてくれるんだ。これを使った次のような命令をステップ8までで作ったプログラムに加えてみよう。

```
215 IF ABS(FX-X)<8 AND ABS(F
Y-Y)<8 THEN 400
400 CUT F,4:BEEP
```

これを加えてRUNさせると、つかまったとたん、ピッ!という音がして、カーソルとOKが表示されるね。これは、400行のあとに

何も命令がないので、プログラムの実行をやめてしまうからだ。

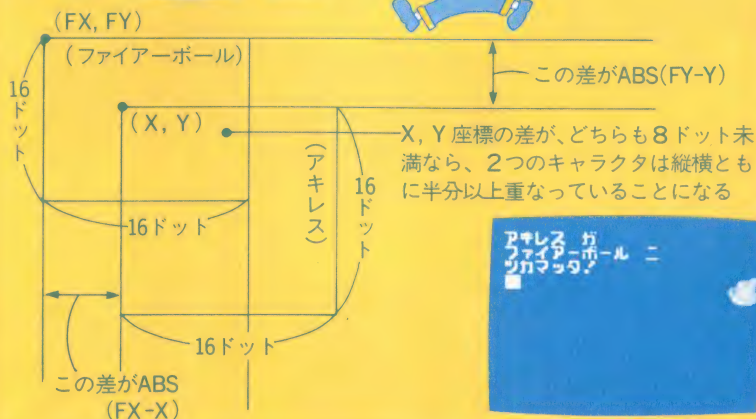
この命令のポイントはIF文のなか。たとえば、ABS(FX-X)は、ファイアーボールとアキレスのX座標の差、ABS(FY-Y)はY座標の差。それがどちらも8(ドット)より小さいということは、縦横ともに半分以上重なっている、つかまったということだ。

```
410 PRINT "アキレス ガ"
420 PRINT "ファイアーボール ニ"
430 PRINT "ツカマッタ!"
440 PAUSE:RUN
```

400行のあとに次の命令を入れておけば、メッセージが出るよ。

PRINT文は4章でも出てきたね。数じゃなくて文字を出したいきは「"」(クォーテーション・マーク) でかこめばいいんだ。

440行では、キーがたたかれたらRUNするように命令している。

図5-9 つかまったかどうかの判断のしかた

▼SAVEとLOAD

苦労したプログラム！テープに保存しておこう！

キーボードとテープレコーダを接続すれば、キミのプログラムをカセットテープに保存できるよ。キーボードのSAVE側の穴とテープレコーダのMICの穴（またはSAVEの穴）、キーボードのLOAD側の穴とテープレコーダのEARの穴（またはLOADの穴）を接続ケーブル（3.5φ ミニジャック。ただし、使えないテープレコーダもあるので注意）でつないで、さあはじめよう。

BG-GRAPHICの使い方

かっこいい背景を作ろうぜ

BG-GRAPICは、かんたんにきれいな背景が作れるお絵描き道具だ。ファミリーベーシックだけにある便利な機能なんだよ。きれいな背景がつくとキミの作ったゲームもグッとひきたつよ。

背景の絵に画面を切りかえるよ

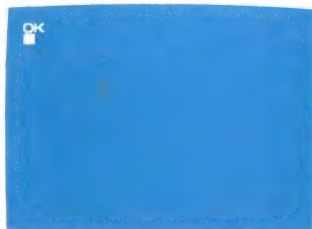
◎SYSTEM命令とESC、STOPの使い方

GAME BASICの初期画面で、2のキーを押せばBG-GRAPHICの画面になるね。ベーシックのプログラムを作っているとちゅうなら、

```
SYSTEM
```

と命令すれば、この初期画面にもどるのでそこで2のキーを押せ

SYSTEMと打ちこんでRETURNキーを押すと初期画面になる

初期画面で2キーを押せばBG-GRAPHICになる

ここで1キーを押すとBASICになる

そこでSTOPキーを押すと初期画面にもどる

ばいいんだ。V3を使っている人はここのやり方がちょっとちがうので、8章を見てね。

逆に、BG-GRAPHICからBASICにもどるには、まず[ESC]を押して、次に[STOP]キーを押せば、初期画面になって、そこで[1]のキーを押せばいいんだ。(図6-1)。

ここで注意！

GAME BASICの初期画面にもどるやり方を忘れてしまったら、ファミコンのRESET（リセットスイッチ）を押して、はじめからやり直せばいいや、なんて考えた人はいないかな。

それもいいかもしれないけど、大切なことを忘れないでほしい。

図6-1 BASICとBG-GRAPHICの切りかえ(V3を除く)

□

X:00 MODE 0
Y:00 SELECT

ここで[ESC]キーを押すと左上に文字が現れる

>SELECT
COPY
MOVE
CLEAR
FILE
CHAR

もし、リセットスイッチを押してしまったら、それまでBASICモードやBG-GRAPHICモードで作っていたプログラムや背景がみんななくなってしまうのだ。もし、どうしてもリセットスイッチを押すのなら、必要なものをSAVEしてからにしよう。

GAME BASICモードで[3]キーを押しても、やっぱり同じようにリセットされるから注意しようね。

さてどこからお絵描きスタート

◎SELECTモードの使い方

BG-GRARHICには、6つのメニューがあるけれど、はじめは、そのうちのひとつ、SELECT（セレクト）になっている。

この面が基本のお絵描きをする面。使えるお絵描き用キャラクタは、キャラクタテーブルBにのってるよ。上から順にAグループ〜Mグループまで，0番から7番のキャラクタがあるね。はじめは、このうちAグループが画面右下に表示されているんだ。

キャラクタの下で点滅しているのがマーカー。スペース・キーを押すと、このマーカーの指しているキャラクタがカーソルの位置にセットされるのだ。マーカーは DEL キーで右へ、INS キーで左へ動くよ。キャラクタグループを切りかえるのは、CLR HOME キーだ。これを押すと、キャラクタグループがAからBグループへ、BからCグループへ移っていく。CからB、BからAへともどすときは、SHIFT キーを押しながら CLR HOME キーだ。

RETURN キーを押すと、配色番号が順々に変わっていく。この配色番号は、バックグラウンド用パレットコードの1の配色番号だ。

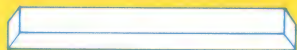
キャラクタのセットされる位置を示すカーソルは、カーソル・キーで動くよ。カーソルの正確な位置を知りたければ、左下のX，Yについている表示を見よう。

図6-2 SELECTモードとキー

カーソル

カーソルの位置に、マーカーが指しているキャラクタがその色でセットされる。

カーソル・キーで上下左右に移動

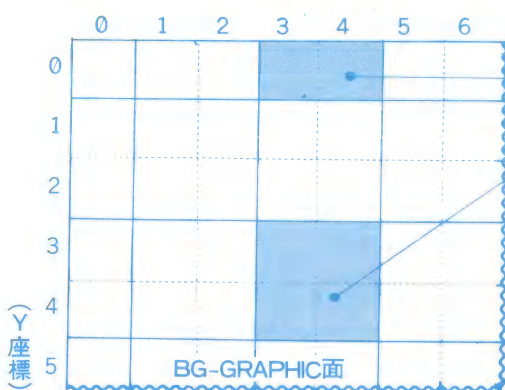
D　Dキーでキャラクタを消す

スペース・キーでキャラクタをセット

背景の絵を作ったり、文字を書きこんだり

◎ファンクション・メニューの使い方

BG-GRAPHICとスプライト面の座標は実はちょっとちがう。でもこれはあとまわしにしておいて、まずはさっそくお絵描きをはじめよう！　もし、すでに画面にセットしたキャラクタを消したいときは、カーソルをその位置に移動させて、Dキーを押せばいいよ。

うまく組みあわさるキャラクタを選んでセットしていけば、思いがけずカッコいい絵になるから、キャラクタテーブルを見ながらじっくりやってみるといいね。かんたんな組みあわせを、あとで紹介しておいたからそちらも参考にしてね。

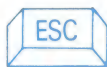
難しいのは配色番号を決めるとき。BG-GRAPHICは4マスご

太線でかこまれたところが1つのカラーエリア。いちばん上だけは2マス分になっているよ。このカラーエリアのなかでは、配色番号が1つしか決められない。

＊注　画面には、この太線や点線は現れないよ。座標の数字で見当をつけよう。

図6-3 ファンクションメニューとモード

ESC を押すと

↓マーカー

>SELECT
COPY
MOVE
CLEAR
FILE
CHAR

画面の左上にファンクション・メニューとマーカーが現れる

カーソル・キーの▼を押すとマーカーは下へ、▲を押すとマーカーは上へ移動する

SELECT
>COPY
MOVE
CLEAR
FILE
CHAR

行きたいモードにマーカーをあわせ、スペース・キーを押すと

スペース・キー

X:00 MODE
Y:00 COPY

画面左下のモード表示が変わって、そのモードになる

とにカラーエリアというのが決まっていて、そのエリアのなかでは、1つの配色番号しか使えないんだ。

*注 CLEARにマーカーをあわせて、スペース・キーを押すと、絵が全部消えてしまうので気をつけて！

たとえば、1つのエリアのなかに配色番号0のキャラクタをおいていたとして、それと同じエリアに配色番号3のキャラクタをセットしたら、前にあったキャラクタも配色番号が3になってしまうのだ。色は、このカラーエリアごとに決めるように考えようね。

でも、この性質を逆に利用して、あとから効率的に色を変えていくこともできるんだ。

さて、SELECTモードのことがわかったら、他のモードもやってみよう。モードを変えるには、ESCキーを押して、カーソル・キーでマーカーを動かし、スペース・キーを押せばいいんだ（図6-3）。モードの切りかえがわかったら、あいだをとばしてCHARモードにしてみよう。このモードは、キーボードから字を打ちこめるモードなんだ。背景に文字を入れたいときに使えるね。

コピーや移動はお絵描きに便利だよ

◎COPYモードとMOVEモード

モードが変わると、カーソル・キーとESCキー（ファンション・メニューを出す）以外は、キーの使い方が変わるので注意。

COPYモードは、SELECTモードで画面にセットしておいたキャラクタをいくつもコピーしてセットできる便利なモードだ。

キーの使い方は図6-4を見てね。

図6-4A
COPYモードでのINS,DELのキーは使い方がちがうので注意

カーソルにキャラクタをコピー

カーソルにコピーしたキャラクタを画面にコピー

F7のキャラクタ　カーソル

INSキーを押すとカーソルにF7がコピーされる

カーソルを移動させるとよくわかる

DELキーを押すと画面にF7をコピー

カーソルを移動すると、F7が残り、カーソルのなかもF7

カーソルにコピーしたキャラクタは、DELキーで好きな位置に何度でもコピーできるけど、消せなくて困らなかったかな？　カーソルのなかのキャラクタを消すには、何もキャラクタのないところでINSキーを押せばいいんだ。つまり背景の黒をコピーしたいということになるんだね。これを応用すると、何もキャラクタのないところでINSキーを押してから、キャラクタにカーソルをかぶせてDELキーを押せば、そのキャラクタを消すことができるよ。

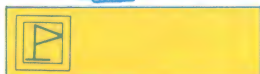
ファンクション・メニューの３番目はMOVEモード。これはCOPYモードとよく似ているけれど、カーソルを動かすともとの場所にキャラクタが残らないという点がちがうところだね。

カーソルにうつしかえたキャラクタは、何もないところでINSキーを押すと、消えてしまうから、注意！

COPYモードとMOVEモードを活用すれば、お絵描きのスピードがぐんとはやくなるよ。

図6-4B MOVEモードでのINS、DELキー

INS

カーソルにキャラクタをうつしかえる

INSキーでカーソルにキャラクタをうつしかえる

カーソルを動かすとキャラクタもついてくる

DEL

カーソルのキャラクタを画面にもどす

DELキーを押すとその場所にキャラクタが残る

画面の消し方残し方

◎CLEARモードとFILEモード

残る2つのモードは、他のモードのように絵や字をかきこむ働きはない。

CLEARモードは、慎重に使おう。今かいていた絵が本当に全部いらなくなったときにだけ、このモードを選ぶのだ。マーカーをCLEARにあわせて、スペース・キーを押したとたん、全部の絵が消えてしまうんだから。一部分だけ消したいときは、SELECTモードのDキーか、CHARモードのスペース・キーで消すこと。

でも、新しい絵をかきたいときはどうしても全部消さなくちゃだめ。それまでの絵をテープに残しておきたいときに使うのがFILEモードだ。

FILEモードにすると、

```
SAVE(S),LOAD(L)?
```

ときいてくるので、テープに保存したいならS、逆にテープに残しておい

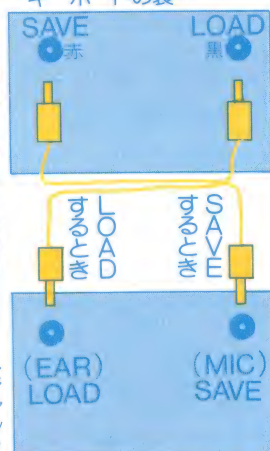
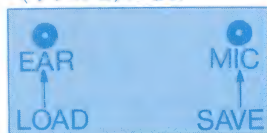
た絵をファミコンにもどしたいときはLを押そう。もちろんその前に、テープレコーダーとキーボードを接続して、テープをセットしておこうね。

どちらを押しても、画面には続けて、

```
SAVE(S),LOAD(L)?S
FILE NAME"■
```

と出てくる。これは残したい絵のタイトルのことだ。好きな名前をキーボードから打ちこもう（ただし16字以内)。最後にBGとつけておけば、プログラムのテープとまちがえずにすむよ。

SAVEのときは、テープが録音をはじめてから、RETURNキー。LOADのときは、RETURNキーを押してからテープをまわすようにしよう。これは、プログラムのSAVEとLOAD(110ページ)のやり方と同じだから、そちらも参考にしてね。

テープレコーダーとキーボードのつなぎ方は図6-5のとおり。ただし、テープレコーダーやラジカセによっては、ファミリーベーシックのプログラムや画面をSAVE、LOADできないものもあるよ。もし、キミの持っているテープレコーダーやラジカセが、ファミリーベーシックとあわないときは、別のもので試してみよう。

こんな絵ができちゃった!

◎キャラクタテーブルBを活用しよう

BG-GRAPHICで絵をかいていくときは、ぜひキャラクタテーブルBを見ながらやっていこう。テレビの画像では気づかなくても、キャラクタテーブルを見ていれば思いがけないキャラクタの組み合わせに気づくよ。

だいたい近くのキャラクタ同士を組み合わせるとうまくいくはずなんだけど、はなれたキャラクタの組み合わせもおもしろいね。

BG-GRAPHICを紙に記録しておくときは、キャラクタテーブルBの記号を使うと便利。本にのるときもこの記号を使って書かれているんだ。たとえば､｢H33｣とあればHグループの3番のキャラクタを配色3で使うということだよ。

	0	1	2	3
0	F40	F50	F50	F50
1		H33		
2		H33		
3		H33		
4				

H 3 3
グループ 番号 配色

(1, 1)の座標に
Hグループの
3番目のキャラクタを
3の配色番号で使う

テーブルとインクツボ(配色番号はすべて0)

				M1	M3	
				M2	M4	
F4	F5	F5	F5	F5	F5	F6
		D3	E2	A5		
			E2			
			E2			
			E2			
		E3	E3	E3		

標識(配色番号はすべて0)

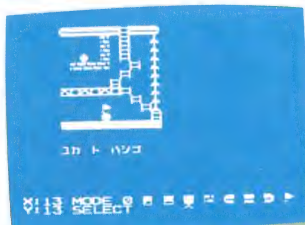
D4	D4			B1	B5	B5	B5	B5	C1
A1	A0	A3		A7	C0	C0	C0	C0	C3
A1	A0	A4	A6	B0	S	T	O	P	C3
A1	A0	A5		A7	B5	B5	B5	B5	C3
A1	A0			B4	C0	C0	C0	C0	D0
D3	A5								
B3	C5								
A7	C3								

解答欄(配色番号はすべて0)

			E6					
D4	E1	E4	E7	E4	E4	E4	E1	D4
D4	E2						E2	D4
D4	E3	E5	E5	E5	E5	E5	E3	D4

いろんな床とハシゴなど

F40	F50	F50	F50	F50	F50	H20	F50	F50	F60
	H33			F33		H20			H41
	H33			F33		H20			H41
	H33	H70		F33		H20			H41
	F33	F33	F33	F33			H20		H41
	H33					H20			H41
	H33				H20				H41
H00	H10	H00	H10	H00	H10	H20			H41
							H20		H41
				F72				H21	H41
				G72					H21
F20	F20	F20	F20	F20	F20	F20	F20	F20	F20

＊左上スミの座標は必ずX，Yとも偶数のところにおくこと

アキレスと背景がいっしょに現れた!

◎VIEW命令と座標の関係

BG-GRAPHICでかいた背景をBASICのなかから呼ぶのが、VIEW命令だ。BG-GRAPHICでお絵描きしたら、ESC + STOPで初期画面へ。そして、BASICにもどり、

```
VIEW
```

と命令してごらん。さっきかいたばかりの絵が現れただろう。この命令は、CLSもかねているので、文字も掃除してくれるのだ。

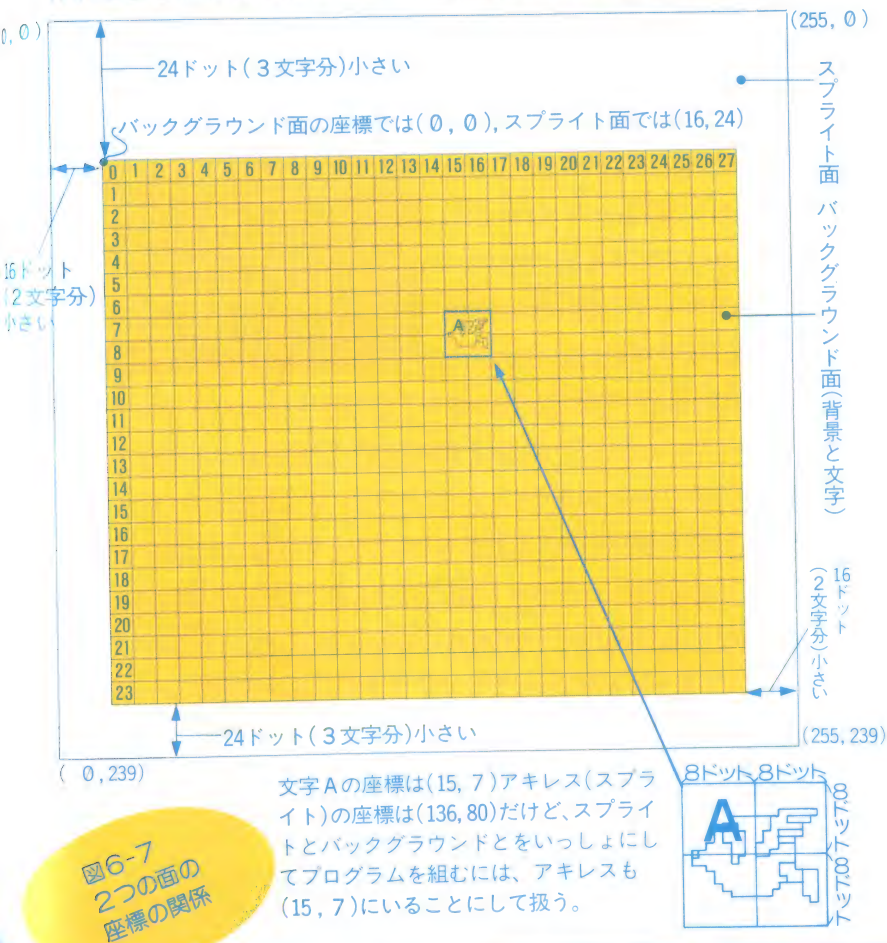
さあ、この命令を使って、ゲーム作りのテクニックを覚えていこう。でも、そのまえに、どうしても知っておかなくてはいけない問題がある。それは、スプライトと背景や文字の座標のちがいだ。

スプライト画はX座標が0～255、Y座標が0～239だけど、背景や文字はX座標が0～27、Y座標が0～23（BG-GRAPHIC面は0～20）なのだ。これは、それぞれ細かさがちがうからだね。

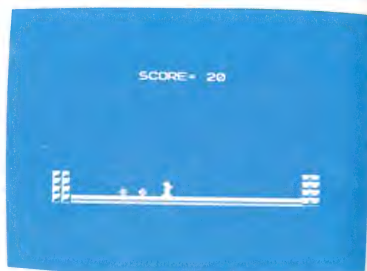
スプライトの座標はドット（点）単位、文字や背景は1文字（8ドット）単位の座標なのだ。それに、スプライト面と文字・背景面（まとめてバックグラウンド面という）とでは

大きさも少しちがうのだ。図6-7を見てね。それに、ふつう動いているアニメ・キャラクタは4文字分（16ドット×16ドット）だったね。このへんがちょっとめんどくさいところだけど、ここさえ乗り切れば背景とキャラクタの動くプログラムをドッキングして、ゲームが作れるようになるんだから、あと1歩だ。がんばってね。

文字Aの座標は(15, 7)アキレス(スプライト)の座標は(136, 80)だけど、スプライトとバックグラウンドとをいっしょにしてプログラムを組むには、アキレスも(15, 7)にいることにして扱う。

図6-7 2つの面の座標の関係

レディがリンゴを取れるんだぞー

◎関数SCR$( )で背景を探知

まず、BG-GRAPHICで、写真のように（0，15）から、(27,15)まで床をかいてね。どんな床でもいいよ。そして、床のすぐ上にリンゴ（Hグループ7番。色は何色でもいい）を適当においてみよう。このリンゴをレディが取っていくプログラムを今から作るんだ。(9，0)～(14，0)には、CHARモードでSCORE＝と書いておいてね。

さて、レディはMOVEで動かすことにして、ちゃんと床の上を歩くようにするには、スプライト面とバックグラウンド面の座標をうまく計算してあわせなくちゃいけないね。そのための計算式が、

スプライトのX座標＝バックグラウンドのX座標×８＋16

スプライトのY座標＝バックグラウンドのY座標×８＋24

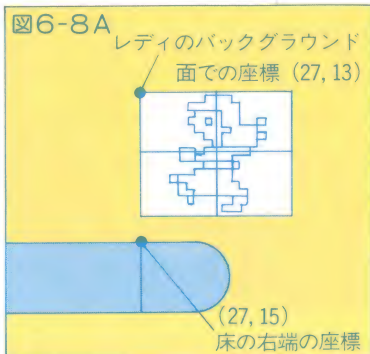
この2つの式だ。これは、図6-7の関係から出てきた式で、とても便利だよ。

レディは画面の右端から歩かせよう。床の右端の座標は、(27，15)だから、レディの座標はその2文字分上、バックグラウンド面の座

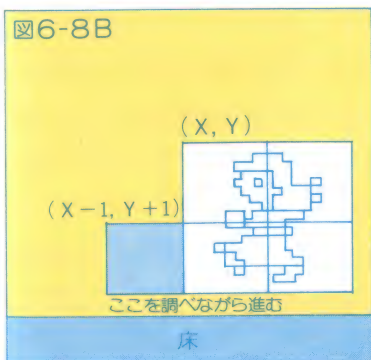
標でいうと（27,13）になるね。（図6-8A）。

そこで、レディのスタート座標をスプライト面に変換すると、

レディのスプライト面でのx座標＝27×8＋16（＝232）

レディのスプライト面でのy座標＝13×8＋24（＝128）

となるね。129ページのプログラムではわかりやすいように、わざと変数の形で書いておいた。

そして、レディのバックグラウンド面での座標を（X，Y）だとすると、レディの進行方向1文字分まえの座標は（X－1，Y＋1）になる。（図6－8B）。この座標にリンゴがきたときに、「取れ」ばいいんだ！　でも、どうやったらわかるのかな？　そこで登場するのが、SCR$(，)という、すごい関数だ。

たとえば、座標（10,10）に"ア"という文字があったとすると、SCR$（10,10）は、まさに"ア"そのものに変わってしまうんだ。もし、そこにリンゴのキャラクタがあれば、SCR$（10,10）はリンゴになっちゃう。

リンゴとりゲームのプログラムだよ

◎関数ASCはCHR$()の反対

ステップ8のSCR$( )関数と組みあわせでよく使うもうひとつの不思議関数にASC( )という関数がある。

コンピュータのなかでは、文字やキャラクタもすべて数字に変えられているということを知ってたかな。スプライトの設定をするときに、CHR$( )という関数を使ったけれど、これはASC( )とは逆に、数字からキャラクタに変換する関数だったのだ。

ファミリーベーシックの取扱説明書に、キャラクタコード表AとBがあるけれど、この表はその数字とキャラクタの関係を教えてくれているんだよ。ただ、スプライト文字や背景とでは数字は同じでもちがった意味になっているのでややこしいかな。

さて、そこで、リンゴ（H7）のキャラクタコード（キャラクタに対応する数字）を探してみよう。キャラクタコード表Bの215(10進数）のところが、H7になってるよ。ということは、リンゴのキャラクタを判定するには、その座標にあるキャラクタSCR$( )を、ASC( )で変換した数字が215かどうかを調べればいいんだ。これがわかれば、次のプログラムが作れるよ。

```
10 VIEW:SPRITE ON:CGSET 1,0
20 DEF MOVE(0)=SPRITE (1,7,2
,4,1,2)
30 X=27:Y=13
40 POSITION 0,X*8+16,Y*8+24
100 A$=SCR$(X-1,Y+1)
110 IF ASC(A$)=215 THEN GOSU
B 200
120 MOVE 0
130 IF MOVE(0)=-1 THEN 130
140 X=X-1
150 IF X-1<0 THEN 30
160 GOTO 100
200 BEEP:SC=SC+10
210 LOCATE X-1,Y+1:PRINT " "

220 LOCATE 15,0:PRINT SC
230 RETURN
```

リスト6—9

ところで、こんなふうにいろいろプログラムを作っていると、そのプログラムがなんのプログラムなのか、あとでパッと見ただけではわからないことがあるね。そこで、REMという命令を使うと便利だよ。これはプログラムのなかにメモするための命令だ。

上のプログラムに、新しい行の命令を、

5 REM リンゴトリ

というふうに、つけくわえてみよう。プログラムの働きはちっとも変わらないんだけれど、これであとで見てもプログラムの内容がわかるね。REMのあとには、自分にわかりやすいようにメモを書くことができるんだ。もし、メモのまえにREMが書いてなかったら、?SN ERRORになっちゃうよ。

今まで、REMを使わなかったのはプログラムを短くして打ちこみやすくするためだったけど、もし、キミがつけたかったら、好きなふうにREMでメモをつけておこうね。

プログラムの終了という意味になるENDや、とちゅうでプログラムの実行を止めるSTOPも今まで使ってないけれど、これらは、もっと長いプログラムを作っていくときに必要になってくるよ。

レディがリンゴ取り ピッ、ピッ、ピッ!

◎GOSUB命令とRETURN

リスト6-9は、10行で画面を呼び出して、20行でレディの動作番号を設定しているのはわかるね。30行〜40行は図6-8Aを見ればわかるはず。レディのスタート位置を決めているんだ。

100行で、いよいよSCR$( )の出番。カッコのなかにある座標の意味は図6-8Bで説明した。この座標にあるキャラクタをA$という変数に移しかえておいて、110行でそのキャラクタが、リンゴかどうか調べているね。140行は、120〜130行でレディが1文字分動くので(20行の「全移動量」が4、つまり8ドットだから)、レディのバックグラウンド面でのX座標を1文字分ずらしているのだ。150行は画面の端まできたときに、はじめにもどる命令。

110行のTHENのあとにあるGOSUB命令は、GOTOに似てい

```
10 VIEW:SPRITE ON:CGSET 1,0
20 DEF MOVE(0)=SPRITE (1,7,2
,4,1,2)
30 X=27:Y=13
40 POSITION 0,X*8+16,Y*8+24
100 A$=SCR$(X-1,Y+1)
110 IF ASC(A$)=215 THEN GOSU
B 200
120 MOVE 0
130 IF MOVE(0)=-1 THEN 130
140 X=X-1
150 IF X-1<0 THEN 30
160 GOTO 100
200 BEEP:SC=SC+10
210 LOCATE X-1,Y+1:PRINT " "
220 LOCATE 15,0:PRINT SC
230 RETURN
```

るけど、ちょっとだけちがう。GOSUB命令である行番号に飛んで実行している途中で、RETURNという命令に出くわすと、GOSUB文のあった場所の次の命令にもどる便利な命令なんだ。ここでは120行にもどっているけど、もし、GOSUB文のあとに：（コロン）で別の命令をくっつけると、その命令から実行するよ。プログラムの改造をするときにも役に立ちそうだね。

200行のSCは"SCORE"を略して変数名にしたもの。こうしておくと、レディがリンゴを取るたびにSCは10ずつ増えていくね。

210行のLOCATEは、はじめて出てきたけど、もうSPRITE命令や、POSITION命令を覚えてきたキミなら、すぐわかるよ。これは、バックグラウンド面でPRINTする位置を、指定できる命令だ。ということは、210行の命令は、リンゴのある位置に空白をPRINTすることになる。つまり、リンゴをこの命令で消しているんだ！

220行はおまけ。上のほうにある "SCORE=" の右どなりにSCの値（つまり、リンゴを取った数×10）を表示しているよ。

さあ、RUNしてみたかな。レディがリンゴを取るたびに、ピッと音がして、リンゴが消えていくね！

レディをあなに落としちゃおか？

◎背景の消し方

今までの命令を応用して、今度は落としあなを作ろう。BG-GRAPHIC面にして、今の床のどこかに2文字分のあなを作ってね。リンゴや“SCORE=”は残しておいてもかまわないよ。

レディが落ちるところは、MOVE命令だと後ろ向きになってつまらないのでSPRITE命令で作ろう。まず、リスト6-9に次の命令を加えてね。

```
25 DEF SPRITE 0,(2,1,1,0,0)=
CHR$(52)+CHR$(53)+CHR$(54)+C
HR$(55)
26 DEF SPRITE 1,(2,1,1,1,0)=
CHR$(53)+CHR$(52)+CHR$(55)+C
HR$(54)
```

今度は、あなを判定するのだから、リンゴのときよりも判定するY座標が1つ下がる（増える）のはわかるね。100行のY+1をY+2に変えればいいんだ。でも、あなのキャラクタって……要するに、空白だね。これもれっきとしたキャラクタで、キャラクタコードは32だよ（キャラクタコード表B）。

すると、110行はこう変わるね。

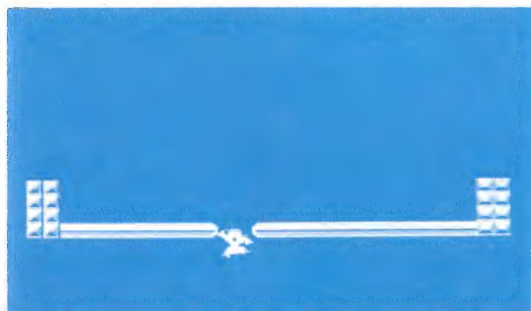
```
110 IF ASC(A$)=32 THEN GOTO
200
```

そして、200行以下をこう変えれば、レディが落としあなに落ちる

プログラムに変わるよ。

```
200 BEEP:ERA 0
210 FOR I=13*8+24 TO 239 STE
P 4
220 SPRITE 0,(X-2)*8+16,I:PA
USE 10:SPRITE 0
230 SPRITE 1,(X-2)*8+16,I+2:
PAUSE 10:SPRITE 1
240 NEXT
250 X=X-3
260 GOTO 40
```

250〜260行は、あなの手前で消した動作番号0を、あなの向こうから動かすための命令だよ。

ほかにも、SCR$（ ）やASC（ ）を応用して、やってみるとおもしろいよ！

いろいろ研究してみよう

BASICのそのほかの言葉

●CGEN……スプライトと文字・背景キャラクタコード表をとりかえてしまう命令。CGEN0なら、どちらもキャラクタコード表A，CGEN1ならふつうとは逆にしてしまう。CGEN2はふつうの設定。CGEN3はどちらもキャラクタコード表Bになってしまう。この命令で、文字をスプライトにしたり、スプライト用のキャラクタを背景に使ったりできるよ。

●POS(0),CSRLIN……それぞれ、カーソルの現在位置のX座標、Y座標を数えてくれる関数の一種。

●COLOR……背景画面に、座標で指定して配色番号を決める命令。これは、BG-GRAPHICのカラーエリアごとに色を決められるよ。たとえば、COLOR1，1，0と命令すると、(0，1)、(1，1)、(0，2)、(1，2) のカラーエリアが配色番号0になるよ。前の2つのパラメータが座標、3つ目が配色番号だ。

●CLEAR……このまま使うと変数や配列をクリアする命令。CLEARのあとにメモリのアドレスを書くと、BASICのプログラムをマシン語のプログラムから守る命令になるよ。

●PEEK, POKE……POKEはメモリの、あるアドレスにデータを書き込む命令。PEEKはメモリの、ある領域からそこにあるデータを取ってくる関数。

●FRE……あと何バイトBASICで使えるかを知るための関数。

●HEX$……10進数の数字を16進数に変える関数。

音楽・計算・エトセトラ

いろいろやってみようぜ!

ファミリーベーシックで、キャラクタを動かしたり、ゲームを作ったりするのも楽しいけど、まだまだほかにもいろんなことができるんだよ。音楽演奏したり、データを自由自在に扱ったり……。この章では、ファミリーベーシックの別の楽しみ方にチャレンジしてみよう。

ファミコンでミュージック♪

☞PLAY文の仕組み

ファミリーベーシックのPLAYという命令は、音楽を演奏する命令だ。PLAYのあとに" "でかこんでファミリーベーシック用の楽譜を書くんだよ。楽譜の書き方はかんたんだ。

こんなふうに、音や休符をアルファベットで表しているんだ。音や休符の長さは、そのアルファベットのすぐあとに数字をつけて決める。

```
PLAY"C5D3E3R7C5D3E3R7"
```

と命令すれば、♩♫𝄽𝄽♩♫𝄽𝄽というリズムになるね。この音の長さの指定は、何も数字をつけなければすぐ前の音と同じ長さになるんだ。だから、今のリズムは、

```
PLAY"C5D3ER7C5D3ER7"
```

と命令しても同じだ。音とアルファベット、音の長さと数字の関

係は表を見てね。これを見ながら、いろいろ実験してみよう。

ファミリーベーシックは、次のようにすれば低いドから高いドまで演奏するよ。

```
PLAY"O3C3DEFGABO4C"                0
```

このなかではO3とかO4がはじめてだね。

これはオクターブを指定しているんだ。Oのあとに0～5の数字をつければ、0ならすごく低い音、5ならすごく高い音が出てくるんだ。Oと数字で指定されたオクターブの高さは次にOと数字で指定しなおされるまで、ずっとその高さで演奏するよ。

図7-1
アルファベットと数字の意味の関係

アルファベット➡音の高さ	
C	ド
#C	ド#(レ♭)
D	レ
#D	レ#(ミ♭)
E	ミ
F	ファ
#F	ファ#(ソ♭)
G	ソ
#G	ソ#(ラ♭)
A	ラ
#A	ラ#(シ♭)
B	シ
R	休み(休符)

数字➡音の長さ	
0	𝅘𝅥𝅰 (𝅀)
1	𝅘𝅥𝅯 (𝄿)
2	𝅘𝅥𝅯. (𝄿.)
3	𝅘𝅥𝅮 (𝄾)
4	𝅘𝅥𝅮. (𝄾.)
5	♩ (𝄽)
6	♩. (𝄽.)
7	𝅗𝅥 (𝄼)
8	𝅗𝅥. (𝄼.)
9	𝅝 (𝄻)

例：C3なら 𝄞𝅘𝅥𝅮 になる

あと、これと同じようにM、Y、V、Tというアルファベットで音の大きさや音符も指定できる。もちろん音の途中で指定しなおすこともできるんだ。

PLAY"O3T4M1Y3V15 A1GA6R3G1FED#C5D5"

オクターブ指定。
O0～O5

テンポ(演奏のはやさ)。
T1～T8。T1ははやく、T8はおそい。

音質を決める
例：M1Y3V15……やわらかく余韻のある音
M1Y0V15……軽く余韻のある音
M1Y3V3……やわらかく歯切れのいい音
ただし、M0だとY(音のやわらかさ)は指定できない。また、V(余韻)も、音の大きさの指定に変わる

メロディー。メロディの途中でもいろんな指定をはさむことができる。このメロディーはドンキーコングで出てきた曲。

カメさんの歩くメロディは?

☞PLAY文をREAD、DATAで活用しよう

実際に他のプログラムにPLAY文を使ってみるとわかるけど、PLAY文を1つ実行しているあいだは他の命令の実行は休んでいるんだ。だから、SPRITE命令などでキャラクタを動かしていると、動きがギクシャクしちゃうのだ。そこで、便利なREAD文とDATA文を使おう。これは、DATAと書かれた行から、1つずつ順番にデータを読みこんでくる命令なのだ。

```
10 FOR I=1 TO 8
20 READ A$
30 PLAY A$
40 NEXT
50 DATA T4M1Y3V15O3C,D,E,F,G
,A,B,O4C
```

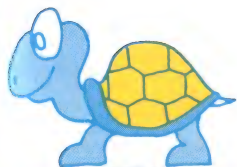
20行と50行で、READ、DATAを使っているね。10、40行のFOR~NEXTで8回、READ A$を繰りかえすたびに、50行のDATAのあとに続く文字を次々に読んで演奏しているのだ。

数字じゃなくて、文字の入る変数を"文字変数"という。文字変数には必ず最後に$マークをつけないと、エラーになるので注意してね。では、これを応用して、カメさんが歩きながらメロディーの流れるプログラムを作ってみよう。

```
10 CLS:SPRITE ON:CGSET 1,2
20 FOR I=0 TO 1
30 DEF SPRITE I,(3,1,0,0,0)=
CHR$(184+I*4)+CHR$(185+I*4)+
CHR$(186+I*4)+CHR$(187+I*4)
40 NEXT
50 PLAY"T103:T103:T103"
100 FOR X=255 TO 0 STEP -4
110 SPRITE 0:SPRITE 1,X,200:
PAUSE 10
120 SPRITE 1:SPRITE 0,X-2,20
0
130 READ A$
140 IF A$="!" THEN RESTORE 2
00:READ A$
150 PLAY A$
160 NEXT
200 DATA G5,E,G:E:E,D,D,R,D,
D,C,D,E,R:R:R,C:E:G,R,G5:G:E
,G,E,D,C,R,D,D,E,D,C,R,C:E:G
,R,!
```

50行のPLAY文は音は出さないけど、テンポとオクターブを決めているんだ。：（コロン）で音のデータを区切ると、3つの音を同時に出せるんだよ。取扱説明書を見て、研究してね。

140行でやっていることは、READとDATAを使うときには大切な命令だ。DATAの最後に〝！〟という文字（記号）があるね。READでデータを読んでいって、A＄がこの文字になったらデータを読む位置を最初にもどして、A＄にデータを入れ直しているんだ。RESTOREという命令は、「はじめからデータの読みこみをはじめろ」という意味なんだ。

このプログラムをRUNすると、カメさんが歩きながらメロディーが流れるよ。

文字変数であそぼう!

☞キーボードから文字を入れるINPUT文

コンピュータは、数だけじゃなくて、文字のたし算もできるんだ。ただし、文字は必ず、" "でくくらないとだめだよ。たとえば、

```
10 CLS
20 A$="オ"+"ハ"+"ヨ"+"ウ"+"!"
30 PRINT A$
```

というプログラムをRUNさせてごらん。画面のうえに、「オハヨウ!」と出てきたね。20行の命令で、文字(" "でかこったもの)を1つ1つたして、その答えを文字変数A$に入れているんだ。画面に出てきたのは、A$の中身だよ(命令は30行)。

こんなふうにできるんだから、いろんな遊びが考えられそうだね。

たとえば、かんたんなうらない遊びを作ってみようか。ここでは、INPUTという命令を新しく使ってみよう。この命令は、キミがキーボードから何か文字(31文字まで)を入れて[RETURN]キーを押すまで、?を出して待つ命令だ。キミの入れた文字は、この命令のあとに書いてある文字変数の中身になるよ。

```
10 CLS
20 INPUT"ナマエハ";N$
30 A=RND(3)
40 LOCATE 7,7:PRINT N$+"サンノウ
ンメイ---":PAUSE 50:BEEP
50 LOCATE 7,8
60 IF A=0 THEN PRINT "サイコウ"
70 IF A=1 THEN PRINT "サイテイ"
80 IF A=2 THEN PRINT "マアマア"
90 PRINT "オワリ!"
```

INPUTと文字変数(N$)のあいだに" " でかこったメッセージを入れて;でつなげば、メッセージに?マークがくっついた形で、表示されるのだ。上のリストをRUNさせると、「ナマエハ?」ときいてくるので、キミの名前を打ちこんで、RETURNキーを押そう。30行でAに入る乱数(でたらめな数)に応じて、キミの運勢を占ってくれる仕組みだ。

INPUTと似た命令に、LINPUTというのもあるよ。これはちょっと変わった使い方ができるので、プログラムのことがよくわかってきたら、いろいろ応用できそうだぞ。

INPUT N$という命令を実行して、コンピュータがデータをきいてきたとき、","(カンマ)を入れると受けつけてくれないんだけど、LINPUT N$としていれば、","だってなんだって、N$のなかにしまいこんでしまうんだ。

それに、もっとおもしろいことがある。INPUTのときと同じように、メッセージもつけられるんだけど、LINPUTのときは、N$のなかにそのメッセージごとはいっちゃうんだ。

たとえば、LINPUT"A=";N$を実行して、1 RETURNとしたとき、N$は何になってると思う? PRINT N$で試してごらん。「A=1」と出てくるよ。

ピピピピピッとメッセージ!

☞ INKEY$とMID$( )

INPUTは命令だけど、関数にも似たような働きをするものがある。それが、INKEY$だ。これは、4章で出てきたSTICK( )やSTRIGとも似ているよ。

だいたい、次のような使い方をすることが多い。

```
10 A$=INKEY$
20 IF A$="" THEN 10
```

こういう命令をプログラムの先頭に持ってきておくと、何かキーを押すまではじまらないようにできるんだ。

10行で、キーボードから入ってきた文字を文字変数A$に入れている。でも、キーボードをさわらなくても、10行の命令は実行されてすぐ次へ行くので、A$には何も入らないことになってしまう。そこで、20行で、もしA$に何の文字も入っていなかったら、また10行へもどるように命令してるんだ。キーボードをさわらないかぎり、このプログラムは、永遠に10行と20行を行ったりきたりしていることになるよ。"" は、文字が何もないという意味なんだ。

もっとも、このプログラムのままでは、何かキーを押してもすぐにプログラムが終わってOKが出るだけ。そこで、次のリストをこれにたしてみるとおもしろいよ。

```
5 CLS
30 LOCATE 5,12
40 FOR I=1 TO 17
50 PRINT MID$("ワタシハ "+CHR$(1
83)+"ミリーコンピュータデス",I,1);
60 BEEP:PAUSE 1
70 NEXT
```

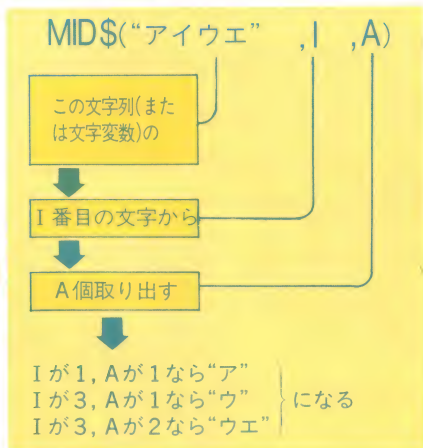
RUNさせると画面が真黒になって、何かキーを押したとたん、"ワタシハ　ファミリーコンピュータデス"と1字ずつビープ音を出しながら表示するよ。これは他にも応用できそうだね。

50行のMID$( )は左から1番目の文字を1つずつ取り出してくれる便利な関数だ（図7-4)。そのなかのCHR$（183）というのは、キャラクターコード表Bから呼び出した"ファ"という文字だ。

ところで、このMID$( )とまえに出てきたRND( )を使ってでたらめなメロディーの自動演奏プログラムがすごくかんたんに作れるよ。

```
10 PLAY MID$("CDEFGAB",RND(7
)+1,1):GOTO 10
```

たったこれだけ！　いろいろ工夫して改造してみてね。

ステップ 5

ファミコンを電卓にしちゃおう

☞ファミリーベーシックの計算機能

やっぱり、コンピュータなんだから計算もさせたいね。もちろん、ファミリーベーシックには、計算機能もあるけど、ふつうの記号とはちがうところがある。今までのプログラムに出てきたように、「たす」「ひく」は「＋」「－」でいいけれど、「かける」は「＊」、「わる」は「／」で計算するようになっているんだ。

ダイレクトモードで計算したいときは、次のようにPRINT文で命令すればいいよ。

```
PRINT 28+56
```

これでRETURNキーを押すと、すぐ下に答えの84を表示してくれる。「かける」だったら、PRINT　28＊56だね。

わり算はちょっとちがうんだ。ファミリーベーシックでは、小数や分数の計算ができなくて、わりきれないときはあまりを切りすてるようになっている。だから、5÷2を計算しようとして、PRINT5／2と命令しても、答えは2としか出してくれない。

そのかわり、あまりを計算してくれる記号もあるんだ。

それがMODという記号。

```
PRINT 5 MOD 2
```

と命令すると、5÷2のあまり、1を表示してくれるのだ。

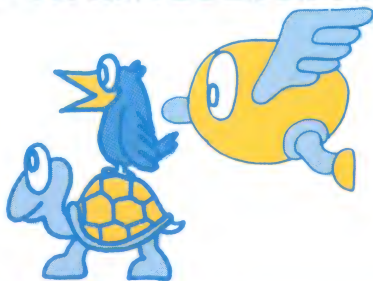
では、INPUT文を今度は数値変数で使って、計算のプログラムを作ってみよう。

```
10 CLS
20 INPUT"2ツノ スウジヲ イレテネ  ";A
30 IF A>99 THEN GOSUB 120:GO
TO 10
40 INPUT"モウ ヒトツ  ",B
50 IF B>99 THEN GOSUB 120:GO
TO 20
60 PRINT "ケイサンシマス":BEEP
70 PRINT A;"-";B;"=";A-B
80 PRINT A;"+";B;"=";A+B
90 PRINT A;CHR$(181);B;"=";A
*B
100 PRINT A;CHR$(182);B;"=";
A/B;"...アマリ";A MOD B
110 END
120 PRINT "オオキスギルヨ!":RETURN
```

RUNさせると「2ツノスウジヲ……」ときいてくるので100以下の数字を、2つ入れよう（100を超えると、「オオキスギルヨ」と出てくるぞ）。キミの入れた2つの数字について、4つの計算結果を出してくれるよ。

70〜100行によく出てくる「；」は数値変数（AやB）と、文字（「ー」や「＋」、それからCHR$は文字変数）をつなぐときにも使うんだ。文字変数同士と数値変数プラス文字変数のときとはちがうので、注意しよう。

CHR$（181）とCHR$（182）はキャラクタテーブルBから、「×」や「÷」の記号を呼び出しているんだよ。

ステップ 6

名前登録のプログラムだよ

☞配列変数とその他の文字関数

いくつかの変数を１つのグループにして、番号で呼び出せるようにできるのが配列変数というものだ。数値変数ならＡ（番号)、文字変数ならA$(番号)のような形で使うのだ。関数に似ているね。

ただ、配列変数を使うまえには、手続きが必要だ。そのための命令が、DIMという命令。ふつう「配列宣言」というよ。

```
DIM A$(10)
```

と命令すれば、A$（0）～A$（10）の配列変数が使えるようになる。DIM Ａ（10）の10は、使いたい配列変数のいちばん大きな番号を入れればいいんだ。

たとえば、10人の名前を文字変数に入れたいとき、いちいち、A$、B$……と10個の変数名を用意するのでは大変。そこで、こんなふうにすれば配列変数が使えるのだ。

PRINT
A$(N)

Ｎを１にすると、
“ヤマダロリ”
Ｎを10にすると
“エイキチ”
が出てくる

0	（未使用）
1	ヤマダロリ
2	ホンダクー
3	ジャマモリ
9	タケチャン
10	エイキチ

```
10 DIM A$(10)
20 FOR I=1 TO 10
30 PRINT I;"バン";
40 INPUT "ナマエ トウロク";A$(I)
50 NEXT
60 BEEP:CLS:INPUT "ナンバン";N
70 PRINT N;"バン ハ  "+A$(N)
80 PAUSE:GOTO 60
```

10行で配列宣言して、20～50行で、1番から10番まで名前が登録できるよ（26文字以内)。1つ終わるごとにRETURNキーを押してね。登録が終わると、ピッと鳴ってCLS。「ナンバン?」ときいてくるので、1～10の数を入れてRETURNキーを押せば、その番号に登録された名前が出てくる仕組みだ。もっと呼び出したかったら、なにかキーを押せば、またきいてくるよ。

もし、名前を全部10文字以内にしたかったら

```
45 A$(I)=LEFT$(A$(I),10)
```

をつけくわえるといいよ。LEFT$(,)は、その文字の左からいくつかだけを取る文字関数なのだ。同じような働きをするものにRIGHT$(,)があるよ。(図7-6B)。

図7-6B LEFT$とRIGHT$

▼こんなこともできるよ!

その他の命令や関数

■キーボードで和音をひこう! ON~GOTOとVAL( )関数

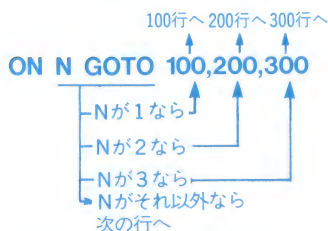
```
10 A$=INKEY$
20 N=VAL(A$)
30 ON N GOTO 100,200,300
40 IF N>3 OR N<1 THEN 10
100 PLAY"C5:E5:G5":GOTO 10
200 PLAY"C5:F5:A5":GOTO 10
300 PLAY"D5:G5:B5":GOTO 10
```

これは、ON~GOTO(図7-7)と、VAL( )(図7-8)という関数を使ったプログラムだ。RUNさせると、1キーを押して"ドミソ"、2で"ドファラ"、3で"レソシ"の和音が出るよ。

INKEY$で入ってきた数字は、数値じゃなくて文字だから、このままではON~GOTOで使えない。そこで、VAL( )が文字を数値に変えてくれるのだ。VAL( )と逆の働きをするものに、STR$( )関数があるよ(図7-8)。

■数字のまえに出るスペースを消そう! LEN( )関数

図7-7 ON~GOTO命令

```
          100行へ 200行へ 300行へ
ON N GOTO 100,200,300
   ├Nが1なら → 100
   ├Nが2なら → 200
   ├Nが3なら → 300
   └Nがそれ以外なら
     次の行へ
```

＊Nのところは式も入れられる

＊GOTOのかわりに、GOSUB、RETURN、RESTOREも使える

PRINT Aなどと、ダイレクトモードで数字を表示させるとよくわかるけど、数字は正の数(0より大きい)の場合、必ず、頭に1つのスペース(空白)があくんだ。マイナスの場合は、そのあきの部分にマイナスの符号がくっつくようになっているんだ。

図7-8 VAL( )とSTR$( )関数

文字を数値に変える

"1 2 3"→ VAL("1 2 3") →1 2 3　　例 VAL("1 2 3")+VAL("1") は124

1 2 3 → STR$(1 2 3) →"1 2 3"　　例 STR$(1 2 3)+STR$(1) は"1231"

数値を文字に変える　　↑PRINT 文でためしてみよう！

これはLOCATE文で、表示位置を指定しても同じこと。その位置から1つ分右にずれちゃうのでややこしくなることがある。

そこで、このスペースをなくす方法を教えちゃおう。ここで出てくるのがLEN( )という関数。これは、文字列の長さ（何文字あるか）を教えてくれる関数だよ。

リストワーク

```
10 INPUT A
20 A$=STR$(A)
30 PRINT RIGHT$(A$,LEN(A$)-1
```

20行で、まず数を数字の文字に変えているのは、次のRIGHT$のなかで使いたいから。30行でやっていることはわかるかな。A$の右側から、A$の文字数より1少ない数だけ取っているんだ。

なぜ？　実は、数を数字の文字に変えるときも、その文字列の頭にスペースが1個くっついているのだ。ためしに、PRINT A$としてごらん。たとえばAに15が入ったとすると、20行でA$の中身は、" 15" となっている。LEN( )は、スペースも1文字と見るので、LEN(A$)は3（文字）になっている。そこで、そこから1ひいて、右側から2文字とれば、スペースなしの15が画面に表示されるというわけ。

この関数は、このプログラムだけだとつまらないけど、あとでキミがゲームを作るときにきっと役に立つよ。

▼リスト7-7をRUNさせて、15と入れてみると…

```
LIST
10 INPUT A
20 A$=STR$(A)
30 PRINT RIGHT$(A$,LEN(A$)-1
)
OK
RUN
?15
15
OK
■
```

■変数や数式の値がプラスかマイナスかわかるSGN(　)関数

SGN(　)という関数は、カッコのなかの変数や式がプラスなら、＋1、マイナスなら−1とこう値になる関数だ、もし、カッコのなかが0なら、SGN(　)も0になるよ。

たとえば、SPRITE命令で追いかけルーチンを作るときに便利だよ。

追いかけるもののX座標をXA、追いかけられるほうのX座標をXBとしたら、XA＝XA−SGN(XA−XB)＊4のようにして、追いかけるもののX座標を計算できるんだ。この式は、XAがXBより大きければ、XAをマイナス4、XAがXBより小さければXAをプラス4するようになっているよ。

V3で拡張された命令・変わったところ

V3のベーシックは強力だぞ!

メモリが2倍になって、内蔵ゲームが4つも入っているファミリーベーシックV3。BASICでプログラムを作りたい！　と思っているキミにとって、便利な命令がいっぱい使えるようになっているぞ。この章では、V3で新しく増えた命令を説明しよう。

2枚のスクリーンが使えるよ

◐BG面0とBG面1のちがい

ファミリーベーシックV3は、はじまるとすぐにBASICになるね。あれ？　BG-GRAPHICは？　なんて心配しなくてもだいじょうぶ。

```
BGTOOL
```

と命令すれば、すぐにおなじみのBG-GRAPHIC面が出てくるんだ。操作方法は5章で説明したのと同じ。

たいせつなことは、このBG-GRAPHIC面が、V3ではBG面1とも呼ばれていることなのだ。もうひとつ、BG面0というのがあって、これは今までのファミリーベーシックでは〝バックグラウンド面〞と呼ばれていたものと同じ。

そして、V3のすごいところは、BASICのSCREENという命令ひとつで、BG面0とBG面1を切りかえられることなのだ。

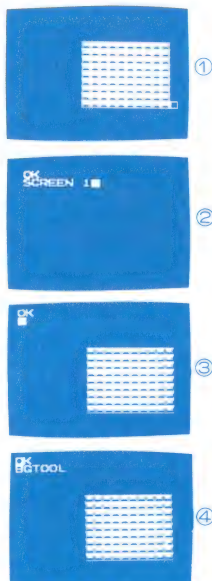
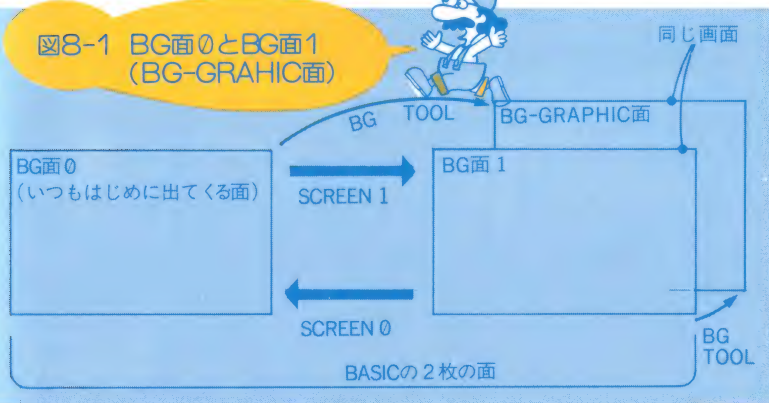
どういうことか、実験してみるね。

まず、BGTOOLと命令して、BG-GRAPHIC面（BG面1）に適当な絵をかいてみよう（写真①）。あとで消すので、かんたんなものでいいよ。

次に、ESC、STOP でBASICにもどり、SCREEN　1と

命令してみよう（写真②）。

　出てきたのは、ＯＫという文字とカーソル、そして、さっきかいたばかりの絵だね（写真③）。VIEW命令なんかしていないのにこの絵が出てきたってことは……この面こそBG面1、つまりBG-GRAPHIC面と同じ面なのだ。ためしに、この面で、BGTOOLと命令してみよう。画面はほとんど変わらないのに、いつのまにか下のほうに、BG-GRAPHICでおなじみの表示が出てきただろう(写真④)。BG面1とBG-GRAPHIC面はこんなふうに、一心同体なのだ。もういちど、ESC と STOP でBASICにもどろう。

　この、いつもはじめに出てくる面はＢＧ面0なのだ。

いま出てるのはどっちの面？

◐SCREEN命令と表示面・アクティブ面

SCREEN命令には、もっとおもしろい使い方がある。それが表示面とアクティブ面の切りかえ指定だ。(図8-2)。

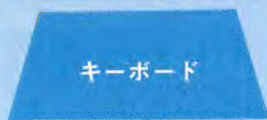
表示面とは、画面に見えている面。アクティブ面とは、PRINT命令やSCR$(　)関数（その面のキャラクタやカラーコードを読む関数）が働きかける面だ。また、キーボードから打ちこむ命令もアクティブ面に書きこまれるよ。

SCREENのあとに、0か1の数字をつけて命令すると、BG面0かBG面1のどちらか一方を表示面兼アクティブ面にすることになるんだけど、これは別々にも指定できるのだ。たとえば、

```
SCREEN 0,1
```

と命令すると、BG面0を表示面に、BG面1をアクティブ面にする。実際にやってみるとわかるけれど、この命令を実行すると、今、キミの見ている面（BG面0）からカーソルが消えて、いくらキーボードをたたいてもなにも出なくなっちゃう。キーボードから打ちこまれた文字は、全部BG面1に行っているからだ。CTRL+Dとやると、カーソルがBG面0にもどってくるよ。

これはいろんな使い方ができておもしろいけど、ひとつだけ、注

意しなくちゃいけないことがある。それは、アクティブ面を1にしたときは、BG面1に文字が入ってしまうということだ。つまり、BG-GRAPHIC面によけいな文字が書きこまれてしまうことがあるんだ。それから、同じようにCLS命令などを、BG面1が、アクティブ面になっているときにやってしまうと、これはBG-GRAPHICでCLEARモードを実行したのと同じことになってしまう。たいせつなBG-GRAPHICを消したり、くずしたりしないように注意しよう！

	(表示面)	(アクティブ面)
SCREEN 0,0 (またはSCREEN 0)	BG面 0　最初の状態	BG面 0
SCREEN 0, 1	BG面 0	＊見えない BG面 1 (BG-GRAPHIC面)
SCREEN 1, 0	BG面 1 (BG-GRAPHIC面	＊見えない BG面 0
SCREEN 1, 1 (またはSCREEN 0)	BG面 1 BG-GRAPHIC面)	BG面 1 (BG-GRAPHIC面)

絵を残すのもかんたん!

プログラムとBG画面をいっしょにセーブ

新しい命令のBACKUPは、今までちょっとむずかしかったメモリのバックアップ（保存）をわかりやすくしたものだ。バックアップは、プログラムを作っているとちゅうで、ちょっと電源を切って休みたいときにそれまで作っていたプログラムを手軽に保存できる機能だね。ただし、V3のカセットにちゃんと単3電池2本を入れておかないと、バックアップできないので注意。

プログラムだけなら、BACKUPと命令するだけでOK。あとは画面に出てくるメッセージどおりに、やればいいんだ。BG画面1（BG-GRAPHIC面）のデータもバックアップしたいときは、BG GETと命令してから、BACKUPとやればいい。この順序をまちがえると正しくバックアップできないよ。

BGGETと命令したあとで、やっぱりBGデータのバックアップ

```
BGGET
OK
BACKUP
```

◀プログラムと画面のバックアップ

はしないことにしたいときは、BGPUTと命令すれば、BACKUPでプログラムだけのバックアップになるよ。

●カセットテープに保存・呼出し

BG画面は、BG-GRAPHIC面のFILEモードからもテープに保存・呼び出しができるけれど、BASICのままでもテープに保存できるようになった。

それがSAVES。反対にLOADするときは、LOADS。この使い方は、プログラムの保存や書きこみに使うSAVE、LOADと同じだよ。

プログラムとBG画面データをいっしょに保存するには（ファイル名“PRO”とすると）、

```
SAVE "PRO" : SAVES "BG-PRO"
```

とするといいね。PROのところはキミの好きなファイル名に変えよう。反対に、いっしょに呼び出すには、

```
LOAD "PRO" : LOADS "BG-PRO"
```

とすればいいよ。要するに、BASICプログラムの命令とBG画面の命令をコロン（：）でつないだだけだね。

ところで、SCREEN命令を応用すれば、BG画面を呼び出している途中経過を見られるよ。呼び出しのときに、SCREEN1, 0：LOADSとしてみよう。画面の3分の1ずつが徐々に現れてきて、色がついて完成。完成したら、CTRL+DでBG面0にもどろう。

動きもグンとゆかいになった

◐CRASH( )とVCT( )の使い方

●カラスがカメをつっついて……CRASH( )のプログラム例

V3では、MOVE命令のなかまにCRASH( )、VCT( )という2つの関数とCANという新しい命令が加わった。どんなふうに使うのかプログラムを作ってみるね。まず、CRASH( )から。次のプログラムを打ちこんでみて。

```
10 CLS:SPRITE ON:CGSET 1,2
20 DEF MOVE(0)=SPRITE (15,7,
3,255,0,0)
30 DEF MOVE(1)=SPRITE (13,7,
1,30,0,3)
40 POSITION 0,240,116
50 POSITION 1,200,120
60 MOVE 0,1
70 IF CRASH(0)=1 THEN PLAY"T
104C0GCG":GOTO 60
80 IF MOVE(0)=-1 THEN 70
```

RUNさせると、カラスがカメさんをつっつくたびに、カメさんが走り出す。これは、70行のIF文で衝突の判定をしているんだ。CRASH（0）という関数は、動作番号0（この場合はカラスさん）

と重なっているスプライトの動作番号が入るんだ。70行は、「カラスさんと衝突したのが、動作番号1（つまりカメさん）なら、音を鳴らしてから、60行へ行け」という意味だ。60行では……動作番号0と1を動かしている。カメさんは、30行で速度を最大にセットしているから、カラスさんに追いたてられるように動くってわけ。

ところで、CRASH( )は、もし2つ以上のキャラクタと重なったら（そのキャラクタが見えなくても）、小さい動作番号のほうが値になるので、使い方に注意しよう！

●マリオがクルクル走るよ……VCT( )のプログラム例

もうひとつVCT( )は、カッコのなかに入っている動作番号のキャラクタがどの方向に設定されているかがわかる関数だ。この関数の値は、そのキャラクタの方向パラメータ（0～8）になる。

```
10 CLS:SPRITE ON:CGSET 1,0
20 H=1:X=30:Y=130
30 DEF MOVE(0)=SPRITE (0,H,1
,30,0,0)
40 POSITION 0,X,Y
50 MOVE 0
60 H=VCT(0)
70 IF MOVE(0)=-1 THEN 70
80 H=(H MOD 8)+1
90 X=XPOS(0):Y=YPOS(0)
100 GOTO 30
```

VCT( )を使ってこんなプログラムを作ってみたよ。RUNすると、マリオがクルクルと回って走るプログラムだ。

●完全に消しちゃうCAN命令

CANという命令は、ERAと使い方は同じだよ。でも、ERAはただ表示しなくなるだけなのに比べて、CANはキャラクタそのものを完全に消す命令なんだ。

プログラム作りもラクチンチン

●行番号をつけるAUTOと整理するRENUM

●行番号を生み出すAUTO命令

```
AUTO
OK
10 ■
```

AUTOと命令してごらん。すぐに、10が出てきてカーソルがその１つあいて右に移動するね。10はコンピュータがつけてくれた行番号なんだ。さっそくなにか打ちこんでみて、RETURNを押してみよう。すぐに、20と出てきて同じようになるね。これは行番号を自動的につけてくれるラクチン命令なのだ。

10、20……だけじゃなく、キミの思いどおりに行番号を作ってもくれるよ。たとえば、

```
AUTO 100,1
```

と命令すると、行番号100から1行ずつ増やして出してくれるのだ。パラメータを入れると最初のパラメータではじまりの行番号、次で何行ずつ増やすかが指定できる。省略したら、コンピュータは、10として考えてくれる。たとえば、AUTO １なら、行番号１から10ずつ、AUTO ，１なら、行番号10から１ずつ増やしてくれる。もう、行番号はいらないというときは、STOPキー。

●行番号を整理するRENUM命令

図8-3 RENUM命令はあっというまに行番号をつけかえてしまう

プログラムを作っていると、とちゅうにいろんな命令を入れたくなって、5とか13とか半端な行番号がいっぱいできちゃうね。これでは見にくいから……と自分でつけかえようとすると、GOTO命令などでエラーが出たりしやすい。そこで、RENUM命令の出番だ。

```
RENUM
```

と命令するだけで、今入っているリストがキチンと10行ごとに整理されてしまうのだ！

●バサッとけずるDELETE命令

プログラムのある行がいらないというときは、その行番号だけ打ってRETURNキーを押せばいいんだけど、たくさんあるとめんどうだね。そこで、DELETE命令を使おう。

たとえば、100行から200行まで全部消したいというとき、

```
DELETE 100-200
```

と命令すれば、一瞬で消えちゃうよ。100行から下は全部いらないというなら、DELETE 100−、100行までがいらないのならDELETE —100と命令すればいいんだ。

でも、うっかり必要な行まで消してしまわないように、使い方には十分注意しようね。

バグとりの秘密兵器だ!

◐流れのわかるTRONと文字を探すFIND

●TRONしたら行番号がぞろぞろ!

BASICの命令がわかるようになってくると、コンピュータがどの行番号の命令をどういう順番で実行しているのかがだいたいわかってくるね。なかなか、思いどおりのプログラムが作れないときは、プログラムリストを目で追って確かめていけば、どこがわるいのかがわかるようになるものなんだ。

でも、やっぱり、見落としてしまったり、あんまり複雑すぎてわからなくなることも多いはず。

そんなときに、試してみたい命令が、TRONだ。なにかのプレグラムを入れておいて、

```
TRON
```

と命令して、OKを確かめてから、RUNしてみよう。画面にいっぱい、#つきの数字を出してきたね。この数字が、今、コンピュータの実行している行番号なのだ。この数字は、あいにく、背景の絵や、LOCATE命令の座標をくるわしてしまうけれど、行番号をよく見ていると、コンピュータがどんな順番で行を実行しているかがわかってくるよ。すると、キミが実行させているつもりの行を全然

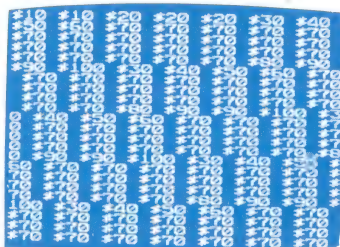
実行していなかったり、行ってほしくないところに行ったりしているのがわかるんだ。

プログラムのまちがいのことを"バグ"(英語で虫という意味)というけれど、このTRONはバグとりの秘密兵器なんだ。TRONをやめたいときには、TROFFと命令すればいいよ。

●FINDでほしい文字を探せ!

たとえば、プログラムを改造したくて、ある変数を探すんだけどなかなか見つからないということもあるね。そんなとき、FINDならかんたんに見つけられるよ。XXという変数を探したいなら、

```
FIND"XX"
```

と命令してみよう。"XX"という文字のある行を全部取り出してくれるぞ。キミは、その行のところだけ見て、好きなように改造すればいいわけだ。この命令は、改造だけじゃなくて、プログラム作りのとちゅうや、バグ取り、それから雑誌や本にのっているプログラムをお手本にしてプログラム作りの勉強にもおおいに役立つぞ。

```
FIND"XX"■
```

```
FIND"XX"
20 X=0:XX=100:Y=0:YY=124
110 IF XX>255 THEN 230
230 XX=XX-255
OK
■
```

ステップ 7

エラーがエラーじゃなくなっちゃった

◐ON ERROR GOTO～とRESUME

この命令は、とても不思議な命令だ。

キミがこの本のプログラムを打ちこんでRUNしてみると、ピッという音とともに“?SN ERROR”とか“?IL ERROR”なんてメッセージが出たことだろう。これが、エラーだ。このエラーの意味はV3のハンドブックのふろくに書いてあるね。

でも、このエラーをプログラムのなかでなくしてしまう命令がある。それが、ON ERROR GOTO～とRESUMEだ。

たとえば、次のプログラムを打ちこんでみよう。

```
10 CLS:SPRITE ON:CGSET 1,0
20 X=120:Y=120:P1=112:P2=116
:PLAY"T1O1M1Y2V5"
30 DEF SPRITE 0,(3,1,0,0,0)=
CHR$(P1)+CHR$(P1+1)+CHR$(P1+
2)+CHR$(P1+3)
40 SPRITE 0,X,Y:PAUSE 10
50 X=X+RND(9)-4:Y=Y+RND(9)-4
:SWAP P1,P2
60 PLAY MID$("C0#FA0#F",RND(
4)*2+1,4):GOTO 30
```

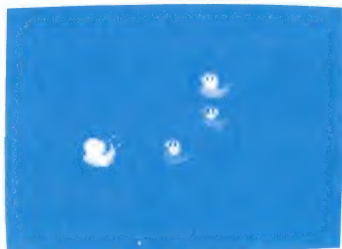
このプログラムは、ファイアーボールを上下左右4ドット以内の範囲ででたらめに動かしているものだ。RUNさせると、“火の玉”らしく、フワフワと画面のまんなかあたりをさまようよ。でも、ず

っと動かしていると、いつかXやYが0～255の範囲から出てしまって、"?IL ERROR IN 40" が出るはずなんだ。

このエラーをふせぐには、4章ステップ7でやったように、IF文を入れればいいんだけど、4行も追加しなくちゃいけない。そこで、次の2行を加えるだけですむのだ。

```
5 ON ERROR GOTO 100
100 X=RND(256):Y=RND(240):RE
SUME
```

5行の命令は、「エラーが出たら80行へ行け」という意味。なんと、この命令は、1度実行すればプログラムのどこで、エラーが発生してもちゃんと100行に行くのだ！そして、100行では、エラーの出ない範囲内にX、Yを適当に決めて、エラーの出たところにもういちどもどるようにしてある。RESUMEという命令が「エラーの出た行にもどれ」という意味だ。XとYはちゃんと範囲内にしてあるから、今度はしばらくエラーが出ないよね。RESUMEには、RESUME NEXT（エラーの出た行の次の行へ）、RESUME〔行番号〕（その行番号へ行け）という使い方もあるよ。

ほかにエラーの関係には、ERL（エラーの発生した行番号がわかる関数）、ERR（エラーコードがわかる）、ERROR〔エラーコード〕（仮にそのコードのエラーを発生させる）があるけど、これは、ON ERROR GOTO～などで作ったエラー処理ルーチンを機能アップしたり、チェックしたりするものだよ。

まだまだこんなにあるよ

V3のその他の命令

●GAME……プログラム中で使うと変なふうになっちゃうけど、ダイレクトで命令すればすぐに内蔵ゲームが楽しめるよ。この命令は F1 ～ F4 キーに入っている。

●FILTER……0～7のカラー番号といっしょに使うと、番号にあわせて画面に色をつけてくれる。FILTER 0は無色、FILTER 1は赤。以下、2は緑、3は黄、4は青、5はマゼンダ、6は空色、7は白。

●CLICK ON/ CLICK OFF……キー入力するときの"パタパタ"という音を出すことにしたり、出さないことにしたりする命令。

●INSTR( )……文字関数の1つ。ある文字列のなかに、もうひとつの文字列がどのあたりに含まれているかを知る関数。たとえば、

```
10  A=INSTR("ABCDE", "DE")
20  PRINT A
```

このプログラムを実行すると、Aの値4が表示されるよ。この数は、"DE"という文字が"ABCDE"のなかの左から4番目に含まれているという意味だ。プログラムでゲームを作っていくうちに、ハイテクニックのひとつとして、他の文字関数(MID$やRIGHT$、LEFT$など)と組みあわせて使うと、威力を発揮してくるぞ。

オリジナル・ゲーム10本

すぐに遊べるプログラム集

この本のために特別にプログラムした楽しいゲームが10本。ベーシックがわかってきたキミには、ゲーム作りの勉強にもなるように、プログラムの解説もしているよ。さっそく、キーボードから打ちこんで、とにかくあそんでみよう！　ゲームは全部、スタートボタンでリプレイできるよ！

ゲーム・プログラム集の遊び方

■まずプログラムリストを打ちこもう！

プログラムリストを1文字ずつよく見ながら、キーボードから打ちこんでいこう。とくに0とO、ⅠとI、Sと$、，と.、;と:などまちがえないように注意。はじめから打っていって、RETURNのところでは、RETURNキーを押して、カーソルが画面の左端に移動したのを確かめてから、次の行番号をから打ちこんでいくこと。

画面の文字は横が28文字で、プログラムリストも横28文字だから、プログラムリストで上下にならんでいる文字は画面でも上下にならぶはずだよ。ときどき、リストと画面の文字の位置があってるかどうか確認するといいね。スペース(あき)の数もキチンとその数だけスペースキーを押すこと。数がわかりにくかったら、その上にならんでいる文字の数を数えればわかるよ。

■BG-GRAPHICデータについて

BG-GRAPHIC面にして、キャラクタテーブルBを見ながら打ちこんでいこう。座標を確認して、たとえば記号がH72ならH(グループの)7(番を)0(のMODE)にしてセットすればいいんだ。6章のステップ6も参考にしてね。また、1、2、6、7、9、10のゲームはBGがなくてもあそべるよ。

本の内容についての問合せは、往復ハガキか60円切手付返信用封筒を同封して、
〒101 東京都千代田区神田錦町3-22
小笠原ビル4F
テクノポリス編集室
ファミリーベーシック入門係
まで。電話の場合は、
☎03-295-4610
まで。
なお、電話による問合せはできるだけ
月曜日〜金曜日の午後5時〜7時
の間にお願いします。

ゲーム1

レディとハエの熱血ファイト
ファイティング・レディ

あそびかた　ハエの動きをよく見てねらおう！

RUNすると、レディが左から右に走っていくよ。でも上空をなんだかじゃまっけなファイターフライがパタパタ飛びまわってるね。そこで、ファイターフライがレディのちょうど真上にきたところで、Aボタンか Bボタン（スタートやセレクトボタンでもいいよ）を押すと、爆発マークが飛びだして、ファイターフライをやっつけるよ。今度はペンペンが動きはじめるから同じようにやっつけよう。次から次にいろんなキャラクタが出てくるよ。でも、1回でもミスったらゲームオーバー。

プログラムはこんなふうにできているよ！

●変数リスト

SC……スコア
X……敵のX座標
Y……敵のY座標
N……動作番号

●プログラムの説明

10～ 50	初期設定
60～ 70	敵の移動
80～100	キー入力
110	レディの移動
120～140	衝突判定
150～170	得点
180～200	ゲームオーバー

ファイティング・レディのプログラムリスト

```
10 CLS:VIEW:CGEN 2:CGSET 1,2
:SPRITE ON [RETURN]
20 Z=0:DIM C(4):C(0)=2:C(1)=
3:C(2)=5:C(3)=15 [RETURN]
3Ø DEF MOVE(Ø)=SPRITE(1,7,1,
4,Ø,2) [RETURN]
4Ø DEF MOVE(1)=SPRITE(1Ø,1,1
,88,Ø,3) [RETURN]
5Ø SC=Ø:X=4Ø:Y=4Ø [RETURN]
6Ø GOSUB 21Ø:POSITION Ø,16,1
7*8+24 [RETURN]
7Ø N=RND(4)+2:POSITION N,X,Y
:MOVE N [RETURN]
8Ø X=XPOS(N):Y=YPOS(N):IF Y>
16Ø THEN Y=24:ERA N:GOTO 7Ø [RETURN]
9Ø MOVE Ø:IF STRIG(Ø)<>Ø THE
N 12Ø [RETURN]
```

ファイティング・レディのBG-GRAPHICデータ

	Ø	1	2	3	4	5	6	7	8	9
Ø		G53								
1										
2										
3		G53			G53					
4										
5										G53
6						G53				
7	G53			GØ3	G13					
8			GØ3	G23	G33	G13				
9	GØ3	G03	G23	G23	G23	G33	G13		G53	
10	GØ3	G23	G23	G23	IØ3	I23	G33	G13		
11	M23	G23	G23	IØ3	I13	I13	I23	G33	G13	GØ3
12	G73	G43	M73	G23	I43	I43	G33	G33	G33	G23
13	M43	G23	G43	IØ3	M53	M53	I23	G33	G33	G43
14	G73	G43	M73	IØ3	I13	I13	I23	G33	G33	G33
15	G23	M73	G43	IØ3	I43	I43	I23	M73	G43	G73
16	G43	M73	G43	G43	I43	I43	H53	M73	M73	M23
17	F43	G43	G43	G43	I43	I43	H53	G43	M73	M43
18	F23	F23	F23	F43	I43	I43	F23	F23	F23	F23
19	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
20	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32

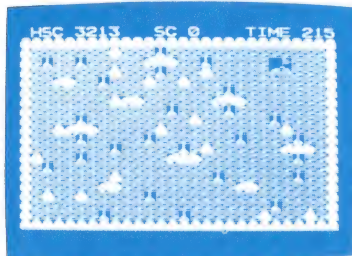
```
100 IF MOVE (N) =-1 THEN 8Ø RETURN
110 ERA N:GOTO 7Ø RETURN
120 POSITION 1, XPOS (Ø), 17*8+
24:MOVE 1 RETURN
130 IF (YPOS (N) -24) /8<> (YPOS
 (1) -24) /8 THEN 13Ø RETURN
140 CUT N:IF ABS (XPOS (1) -XPO
S (N) ) >=16 THEN 18Ø RETURN
150 SC=SC+5:PLAY"T1O3CDE" RETURN
160 IF MOVE (1) =-1 THEN 16Ø RETURN
170 ERA 1, N:Z= (Z+1) MOD 4:GO
TO 6Ø RETURN
180 LOCATE 8, 12:PRINT"YOUR S
CORE";SC:PLAY"T2OØD3R3D6" RETURN
190 IF STRIG (Ø) =1 THEN RUN RETURN
200 GOTO 19Ø RETURN
210 FOR I=1 TO 4:DEF MOVE (I+
1) =SPRITE (C (Z), I*2, 1, 2Ø, Ø, 1)
:NEXT:RETURN RETURN
```

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
				G53						G53				G53			
														G53			G53
	G53						G53										
										G53	GØ3	G13			G53	G53	
				G53						GØ3	G43	G43	G13				
								G53	GØ3	G43	G43	G43	G43	G13	G53		
								GØ3	G43	M73	G23	G33	M73	G43	G13		
	G53	G53			G53		GØ3	G43	G23	G23	G43	G43	G33	G33	G43	G13	
						GØ3	G43	M73	G23	M73	G43	G43	G33	G33	G33	G43	G13
					GØ3	G43	G23	G23	M73	G43	M73	G43	M73	G33	G33	M73	G43
GØ3	G13	I33	I33	GØ3	G43	G23	M73	G23	G43	M73	G43	M73	G43	G43	G33	G33	G33
G43	G33	I13	I13	I23	G23	G23	G23	G43	M73	I43	I33	G43	M73	G43	M73	G33	G33
G23	G33	I13	I13	G43	M73	M73	G43	M73	M73	GØ3	I13	I23	G43	M73	G43	M73	G43
G43	G43	M73	G43	M73	G33	G43	G43	M73	G33	I43	I43	G43	M73	G43	G43	G43	G33
M73	G43	M73	G43	G43	G43	M73	G43	G43	G43	IØ3	I13	I23	G43	G33	M73	G43	G43
G43	M73	G43	G43	G33	G43	G33	G43	G33	G43	GØ3	I13	I23	G43	G33	G43	M73	G43
M73	G43	M73	M73	G43	M73	G43	M73	G43	G43	G43	I43	M53	G43	G43	M73	G33	G43
G43	G43	M73	G43	G43	G43	G33	G43	G43	M73	G43	I43	M53	M73	G43	G33	M73	G43
F23	F23	F23	F23	F23	F23	F23	F23	F23	F23	F23	I43	F23	F23	F23	F23	F23	F23
F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F33	F32	F32	F32	F32	F32	F32
F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32

●原作・もりあきら『マジックハンド』(プログラムポシェットVOL.3)より移植

ゲーム2

あれれ？ どっちへ行くの？！
不思議の森のピクニック

あそびかた

毒シダの近くで止まらないように！

ここは不思議(ふしぎ)の森(もり)。とにかくおかしなことばかりおこるんだ。

キミはボールになって、旗(はた)のところまで行(い)こうとするんだけれど、✚ボタンを押(お)すとおかしな方向(ほうこう)に行(い)ってしまうのだ。そう、この森(もり)では、ピッと音(おと)がするたびに✚ボタンの方向(ほうこう)がおかしくなってしまうんだ。だからときどき方向(ほうこう)を確(たし)

不思議の森のピクニックのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0										
1	130	130	130	130	130	130	130	130	130	130
2	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
3	130	G40	H40	G40	G40	H40	G40	G40	G40	G40
4	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
5	130	G40	G40	G40	G40	G40	G40	H40	G40	G40
6	130	G40	G40	100	120	G40	G40	G40	G40	G40
7	130	G40	G40	G40	G40	G40	G40	G40	G40	100
8	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
9	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
10	130	G40	G40	G40	G40	H40	G40	G40	G40	G40
11	130	G40	G40	G40	G40	110	120	G40	G40	G40
12	130	G40	G40	G40	G40	H40	G40	G40	H40	G40
13	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
14	130	G40	G40	G40	G40	H40	G40	G40	G40	G40
15	130	G40	H40	G40	G40	G40	G40	G40	G40	G40
16	130	G40	G40	G40	G40	G40	G40	G40	120	G40
17	130	G40	G40	G40	G40	G40	G40	100	G40	G40
18	130	G40	G40	G40	G40	G40	G40	G40	G40	G40
19	130	G40	G40	G40	G40	G40	G40	H40	G40	G40
20	130	130	130	130	130	130	130	130	130	130

プログラムはこんなふうにできているよ！

●変数リスト

A,B,C,D……キー方向
X,Y……現在位置
T……時間
S……スコア
H……ハイスコア
E……X方向移動ベクトル
F……Y方向移動ベクトル

●プログラムの説明

10～ 90　初期設定
100～120　キー方向決定
130～200　移動および衝突判定
210～250　ゲームオーバー判定

かめながら、音がしないうちに早く移動してしまわないと、とても目的の旗までたどりつけないよ。

　それに、この森はところどころに毒シダがはえていて、これにぶつかるとゲームオーバーになってしまう。旗はすぐ近くにありそうだけど、なかなかたどりつけないぞ。

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130
G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	100	110	120	G40	G40	H40	G40	G40	G40	G40	G40		F73	G40	G40	G40	130
G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	G40	103	123	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	130
120	G40	G40	G40	G40	G40	G40	100	110	120	G40	G40	G40	G40	G40	G40	G40	130
G40	G40	G40	H40	G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	H40	G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	H40	G40	G40	130
G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	100	110	120	G40	130
G40	G40	G40	100	110	120	G40	G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	H40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	H40	G40	G40	130
G40	H40	G40	G40	G40	G40	G40	H40	G40	100	120	G40	G40	G40	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
G40	G40	G40	G40	G40	G40	G40	H40	G40	G40	G40	G40	H40	G40	G40	G40	G40	130
G40	G40	G40	G40	H40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	G40	130
130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130

不思議の森のピクニックのプログラムリスト

```
10 KEY 1, "POKE &H7800, 1, 0"+C
HR$ (13) RETURN
20 CGEN 3:SPRITE ON:CLS:CGSE
T 1, 1:PALET S 3, 0, 48, 22, 22 RETURN
30 DEF SPRITE 1, (3, 0, 0, 0, 0) =
CHR$ (215) RETURN
40 A=2:B=1:C=8:D=4 RETURN
50 X=3:Y=18:T=300:L=L+20:CLS
:VIEW:H=PEEK (&H7800) *100+PEE
K (&H7801) RETURN
60 FOR N=1 TO L RETURN
70 LOCATE RND (25) +1, RND (18) +
2:PRINT CHR$ (219) :NEXT RETURN
80 LOCATE 23, 3:PRINT CHR$ (19
9) RETURN
90 LOCATE 1, 0:PRINT"HSC";H;"
   SC";S RETURN
100 BEEP:K=RND (3) :IF K=0 THE
N SWAP A, B:SWAP C, D RETURN
110 IF K=1 THEN SWAP A, C:SWA
P B, D RETURN
120 IF K=2 THEN SWAP A, D:SWA
P B, C RETURN
130 FOR M=1 TO 40:I=STICK (0)
:T=T-1 RETURN
140 LOCATE 20, 0:PRINT"TIME";
T RETURN
150 IF T=0 THEN 210 RETURN
160 E= (I=A) - (I=B) :F= (I=C) - (I
=D) RETURN
170 LOCATE X, Y:PRINT CHR$ (20
4) :X=X+E:Y=Y+F RETURN
```

```
180 P$=SCR$(X,Y):IF P$=CHR$(
199) OR P$=CHR$(219) THEN 20
0 RETURN
190 SPRITE 1,(X*8)+16,(Y*8)+
24:NEXT:GOTO 100 RETURN
200 IF P$=CHR$(199) THEN S=S
+T:GOTO 50 RETURN
210 L=0:LOCATE 10,12:IF S>H
THEN H=S:POKE &H7800,H/100,H
 MOD 100 RETURN
220 S=0:PRINT"ジ.END":SPRITE
1 RETURN
230 LOCATE 3,14:PRINT"TRY AG
AIN (PUSH 'S' KEY)" RETURN
240 IF STRIG(0)=1 THEN RUN RETURN
250 GOTO 240 RETURN
```

＊F1キーを押すとハイスコアは100にもどります

●原作・山本浩『絶命・入らずの森』(プログラムポシェットVOL・3)より移植

ゲーム3

ファイターフライ軍団の襲撃だ

ハエ・ハエ・カカカ！

あそびかた

エアーポケットを探し出せ！

白い雲のうかぶ青空にマリオがポッカリ浮かんでいるね。でも、ファイターフライ軍団が左右からワーッとおそいかかってくるぞ。✚ボタンでマリオを動かして、ファイターフライにぶつからないように逃げまわろう。

マリオはとてもじょうぶなので、ファイターフライがかすったく・

ハエ・ハエ・カカカ！のBG-GRAPHICデータ

プログラムはこんなふうにできているよ！

●変数リスト

X( )……ハエのX座標
Y( )……ハエのY座標
XX……マリオのX座標
YY……マリオのY座標
TM……時間
N……入力キー
HS……ハイスコア

●プログラムの説明

10～ 90	初期設定
100～110	ハエの位置を決めて動かす
120～170	キー入力
180～220	座標の変更とマリオの表示
230～260	衝突判断
270～320	ゲームオーバー判定

らいじゃビクともしないけど、もろにぶつかってしまったら、とたんにゲームオーバーだ。

画面(がめん)の上(うえ)でクルクルッとふえている数字(すうじ)が、キミのにげまわりタイム。このタイムがキミの得点(とくてん)になるよ。

ときどき、パッとワープしておそってくるから気(き)をつけてね。

ハエ・ハエ・カカカ！のプログラムリスト

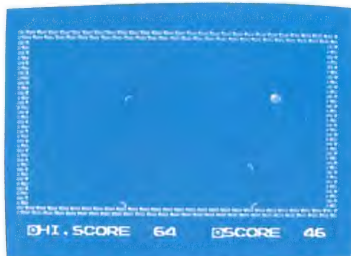
```
10 CLS:VIEW:CGEN 2:CGSET 0,1
:SPRITE ON:PALET B 0,1,48,1,
1 RETURN
20 DIM X(8),Y(8) RETURN
30 FOR I=1 TO 7:DEF MOVE(I)=
SPRITE(2,4+(I MOD 2)*2,1,200
,0,RND(4)):NEXT RETURN
40 DEF SPRITE 1,(3,1,0,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3) RETURN
50 DEF SPRITE 2,(3,1,0,0,0)=
CHR$(8)+CHR$(9)+CHR$(10)+CHR
$(11) RETURN
60 DEF SPRITE 3,(3,1,0,1,0)=
CHR$(1)+CHR$(0)+CHR$(3)+CHR$
(2) RETURN
70 DEF SPRITE 4,(3,1,0,1,0)=
CHR$(9)+CHR$(8)+CHR$(11)+CHR
$(10) RETURN
80 XX=120:YY=120:TM=0:M=1 RETURN
90 HS=PEEK(&H783A)*256+PEEK(
&H783B) RETURN
100 FOR I=1 TO 7:X(I)=RND(27
)*8+16:Y(I)=RND(21)*8+24:POS
ITION I,X(I),Y(I):NEXT:I=1 RETURN
110 MOVE 1,2,3,4,5,6,7 RETURN
120 TM=TM+1:N=STICK(0):IF N<
>0 THEN SPRITE M:M=M MOD 2 RETURN
130 LOCATE 5,0:PRINT"TIME";T
M RETURN
140 IF N=0 THEN 190 RETURN
150 IF N=1 THEN XX=XX+4:M=M+
3:GOTO 190 RETURN
```

```
160 IF N=2 THEN XX=XX-4:M=M+
1:GOTO 190 RETURN
170 IF N=4 THEN YY=YY+4:M=M+
3:GOTO 190 RETURN
180 IF N=8 THEN YY=YY-4:M=M+
1 RETURN
190 IF XX>240 THEN XX=5:GOTO
 230 RETURN
200 IF XX<6 THEN XX=240:GOTO
 230 RETURN
210 IF YY>220 THEN YY=5:GOTO
 230 RETURN
220 IF YY<6 THEN YY=220 RETURN
230 SPRITE M, XX, YY:PLAY"T1M1
Y0O1C2" RETURN
240 IF ABS(XX-XPOS(I))<16 AN
D ABS(YY-YPOS(I))<16 THEN 28
0 RETURN
250 I=I+1:IF I>7 THEN GOTO 1
00 RETURN
260 IF MOVE(I)=-1 THEN GOTO
120 RETURN
270 GOTO 100 RETURN
280 BEEP:LOCATE 9, 8:PRINT"GA
ME OVER" RETURN
290 LOCATE 9, 10:PRINT"TIME";
TM RETURN
300 IF TM>HS THEN HS=TM:POKE
 &H783A, HS/256, HS MOD 256 RETURN
310 LOCATE 7, 12:PRINT"BEST T
IME";HS RETURN
320 LOCATE 8, 14:PRINT"TRY AG
AIN ?":IF STRIG(0)=1 THEN RU
N RETURN
330 GOTO 320 RETURN
```

●原作・横田啓『BEES!』(プログラムポシェットVOL・3)より移植

ゲーム4

キミはなんかいできるかな?
バウンド・ボール

あそびかた

ボールの道すじをはじめに計画!

RUNさせたら、Aボタンでアマチュア用、Bボタンでプロ用のバウンド・ボールが選べるよ。AかBを押してスタートボタンでゲームスタート。ボールが飛びはじめたら、Aボタンで″＼″、Bボタンで″／″が出現してボールをはねかえすよ。これでボールが壁にぶつからないように飛ばしていると

バウンド・ボールのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
1	F32									
2	F32									
3	F32									
4	F32									
5	F32									
6	F32									
7	F32									
8	F32									
9	F32									
10	F32									
11	F32									
12	F32									
13	F32									
14	F32									
15	F32									
16	F32									
17	F32									
18	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
19										
20		D40	H	I	.	S	C	O	R	E

プログラムはこんなふうにできているよ！

●変数リスト

P,Q……ボール移動方向を決める

HI……ハイスコア

S……得点

X,Y……ボールの座標

A,B……入力キーチェック用

●プログラムの説明

10〜 80	初期設定
90〜120	スタート表示
130	ボールの方向、座標初期設定
140	空白プリント
150	キー入力
160〜170	入力キー判断
180	A―B―A―B……と10回くり返しキー
190	得点表示
200	ボール表示
220	ゲームオーバー
230	ハイスコア比較
240〜260	リプレー
270〜280	ボール移動

スコアがどんどんあがる仕組み。ただし、ＡＢＡＢ……と10回繰りかえすとゲームオーバーになっちゃうから、どこかとちゅうでＡＡとかＢＢと同じキーを２回押してね。

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
																	F32
F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
							D42	S	C	O	R	E					

バウンド・ボールのプログラムリスト

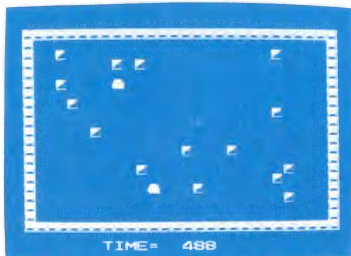
```
10 CLS:LOCATE 1,10:PRINT"LEV
EL ? (A)->AMA (B)->PRO" RETURN
20 T=STRIG(0):IF T<4 THEN 20
 RETURN
30 IF T=8 THEN M=10:GOTO 50 RETURN
40 M=0 RETURN
50 VIEW:CGEN 2:CGSET 1,1:PAL
ETB 0,47,48,39,22 RETURN
60 DIM P(3),Q(3):P(1)=1:P(3)
=-1:Q(0)=-1:Q(2)=1 RETURN
70 HI=PEEK(&H783A)*100+PEEK(
&H783B) RETURN
80 LOCATE 11,20:PRINT HI RETURN
90 LOCATE 7,8:PRINT"PUSH (ST
ART)" RETURN
100 T=STRIG(0):IF T<>1 THEN
100 RETURN
110 LOCATE 7,8:PRINT"
       " RETURN
120 LOCATE 24,20:PRINT S RETURN
130 D=RND(4):X=RND(10)+8:Y=R
ND(7)+7 RETURN
140 LOCATE X,Y:PRINT" " RETURN
150 T=STRIG(0):IF T<4 THEN G
OSUB 270:GOTO 190 RETURN
160 IF T=8 THEN GOSUB 270:LO
CATE X,Y:PRINT CHR$(240):A=A
+1:B=0:D=3-D:GOSUB 270:GOTO
180 RETURN
```

```
170 IF T=4 THEN GOSUB 270:LO
CATE X,Y:PRINT CHR$(239):A=0
:B=B+1:D=(5-D) MOD 4:GOSUB 2
70 RETURN
180 SWAP A,B:PLAY"O4F1D":IF
 (A+B)>9 THEN 220 RETURN
190 S=S+1:LOCATE 24,20:PRINT
 S RETURN
200 LOCATE X,Y:PRINT CHR$(20
7) RETURN
210 PAUSE M:GOTO 140 RETURN
220 LOCATE 10,8:PRINT"GAME O
VER":PLAY"O1C1BC5" RETURN
230 IF HI<S THEN POKE &H783A
,S/100,S MOD 100 RETURN
240 LOCATE 10,10:PRINT"REPLA
Y ?" RETURN
250 TR=STRIG(0):IF TR<>1 THE
N 250 RETURN
260 RUN RETURN
270 X=X+P(D):Y=Y+Q(D):IF SCR
$(X,Y)<>" " THEN 220 RETURN
280 RETURN RETURN
```

●原作・児島冬樹『BOUND BALL』プログラムポシェットVOL.1より移植

ゲーム5

動かしかたがメチャむずかしい
へんてこブルドーザー

あそびかた　四角をかくように操作しよう！

操縦がとてもむずかしい、十字型のへんてこブルドーザー。コントローラのAボタン、Bボタンで操縦して、四角い鉱石を取っていこう。ただし、水色のインベーダーやまわりのかべにあたっちゃったら、ゲームオーバーだよ。下に表示されているTIMEが0になってもゲームオーバー。

へんてこブルドーザーのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01
1	F01									
2	F01									
3	F01									
4	F01									
5	F01									
6	F01									
7	F01									
8	F01									
9	F01									
10	F01									
11	F01									
12	F01									
13	F01									
14	F01									
15	F01									
16	F01									
17	F01									
18	F01									
19	F01									
20	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01

プログラムはこんなふうにできているよ！

●変数リスト

H……面数

X,Y……ブルドーザー座標

A……鉱石をとった数

T……TIME

K……地雷の数

S……スコア

●プログラムの説明

10～ 20	初期設定
30～ 40	画面設定
50～ 80	キー入力処理
90～150	移動先にどんなキャラクターがあるか？
160	ブルドーザー表示
170	面クリア処理
180～200	ゲームオーバー処理
210	乱数を作る

さて、操縦法を教えるね。Aボタンを１回だけ押すと、ブルドーザーが右に進んでいたら左に、左に進んでいたら右に向きをかえる。Bボタンは、押しつづけているあいだ、進行方向に対して右に進むんだ。つまり、右に進んでいるときは下へ、左に進んでいるときは上に行くのだ。頭がこんがらがってきたところで、ゲーム開始！

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
																	F01
F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01	F01

へんてこブルドーザーのプログラムリスト

```
10 CGEN 2:CGSET 0,2:SPRITE O
N:M=1 RETURN
20 X=19:Y=9:W=9:A=10:T=550-M
*50:CLS:VIEW RETURN
30 L=1:K=M+1:LOCATE 7,22:PRI
NT"TIME= " RETURN
40 FOR I=1 TO 15:GOSUB 210:L
OCATE C,D:PRINT CHR$(192):NE
XT:FOR I=1 TO K:GOSUB 210:LO
CATE C,D:PRINT CHR$(219):NEX
T RETURN
50 I=STRIG(0):IF I<>0 THEN P
LAY"T1M1Y3O1C" RETURN
60 IF I=4 THEN W=Y+L:GOTO 80
 RETURN
70 V=X+L RETURN
80 IF I=8 THEN L=-L RETURN
90 P=ASC(SCR$(V,W)) RETURN
100 IF P<>192 THEN 130 RETURN
110 BEEP:S=S+10:A=A-1:IF A=0
 THEN 170 RETURN
120 IF RND(9)=1 THEN GOSUB 2
10:LOCATE C,D:PRINT CHR$(199
) RETURN
130 IF P=219 OR P=193 THEN 1
80 RETURN
140 T=T-1:LOCATE 12,22:PRINT
" ";T;" ";:IF T=0 THEN 180 RETURN
150 IF P=199 THEN S=S+50:PLA
Y"T1M1Y2O5G" RETURN
```

```
160 LOCATE X,Y:PRINT"  ";:X=V
:Y=W:LOCATE X,Y:PRINT CHR$(2
32):GOTO 50 [RETURN]
170 B=T/2:LOCATE 10,8:PRINT"
BONUS ";B:S=S+B:LOCATE 10,12
:PRINT"SCORE=";S:FOR I=1 TO
9000:NEXT:M=M+1:GOTO 20 [RETURN]
180 PLAY"O4B1AO3GFEDCO1C5":L
OCATE 7,8:PRINT"    GAME OVER
   ";:LOCATE 10,13:PRINT"SCO
RE=";S:HS=PEEK(&H783A)*256+P
EEK(&H783B):IF HS<S THEN HS=
S [RETURN]
190 LOCATE 9,15:PRINT"HI-SCO
RE=";HS:POKE &H783A,HS/256,H
S MOD 256:IF STRIG(0)<>1 THE
N 190 [RETURN]
200 RUN [RETURN]
210 C=RND(24)+2:D=RND(16)+2:
RETURN [RETURN]
```

●原作・藤田康二『まる対さんかく』(プログラムポシェット85・NO・1)より移植

ゲーム6

なんだかのんびり戦闘ゲーム
ニットピッカーvs.ファイターフライ

あそびかた　のんびりムードにだまされないで！

今日はいいお天気なので、ファイターフライのキミは空のお散歩を楽しんでいるところ。ところがこわいニットピッカーたちがゾロゾロ……。つかまったら、パックり食われちゃうぞ！

ファイターフライは、キミがなにもしなくても、空からフラフラストーンと降りてきて、地面につ

ニットピッカーvs.ファイターフライのBG-GRAPHICデータ

	∅	1	2	3	4	5	6	7	8	9
∅										
2										
3										
4										
5										
6										
7										
8			I03	I23						
9			I03	I23						
10		I03	I13	I13	I23					
11		I03	I13	I13	I23			I33		
12	J03	I13	I13	I13	I03	I23	I03	I13	I23	
13	I03	I13	I13	I13	I03	I23	I03	I13	I23	I03
14	I13	I13	I13	I13	I03	I03	I13	I13	I13	I23
15			H51	H51		I03	I13	I13	I13	I23
16			H51	H51		I03	I13	I13	I13	I23
17			H51	H51				H51		
18			H51	H51				H51		
19			H51	H51				H51		
20	G31	G41	G21	G31	G41	G31	G21	G21	G31	G41
1	G21	G31	G41	G41	G31	G41	G21	G31	G41	G21

プログラムはこんなふうにできているよ！

●変数リスト

K……キー入力

X,Y……ファイターフライの座標

W……ファイターフライの移動量

S……スコア

X( ),Y( )……ニットピッカーの座標

HS……ハイスコア

●プログラムの説明

10～ 40	初期設定
50	キー入力処理
70～100	ニットピッカー移動・表示
110	ファイターフライ表示
130	ニットピッカーとファイターフライが衝突したか
150～190	ゲームオーバー処理

いたら、またピューンと空(そら)へ舞(ま)いがるよ。そこで、キミはニットピッカーに食(く)われないように✚ボタンで左右(さゆう)に動(うご)かそうね。ちょっとでも、ニットピッカーにさわるとゲームオーバーになっちゃうからご用心!!

スコアは、右端(みぎはし)からニットピッカーの出(で)てきた数(かず)。

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
			I33														
		I03	I13	I23													
		I03	I13	I23													
		I03	I13	I23													
	I03	I13	I13	I13	I23				I03	I23					I03	I23	
	I03	I13	I13	I13	I23				I03	I23					I03	I23	
I03	I13	I13	I13	I13	I13	I23		I03	I13	I13	I23			I03	I13	I13	I23
I03	I13	I13	I13	I13	I13	I23		I03	I13	I13	I23			I03	I13	I13	I23
I03	I13	I13	I13	I13	I13	I23	I03	I13	I13	I13	I13	I23		I03	I13	I13	I23
I13	I13	I13	I13	I13	I13	I13	I23	I13	I13	I13	I13	I23	I03	I13	I13	I13	I13
I13	I13	I13	I13	I13	I13	I13	I23	I13	I13	I13	I13	I23	I03	I13	I13	I13	I13
I13	I13	I13	I13	I13	I13	I13	I13	I23	I13	I13	I13	I13	I23		H51	H51	
		H51	H51	H51		I03	I13	I13	I13	I13	I13	I13	I23		H51	H51	
		H51	H51	H51		I03	I13	I13	I13	I13	I13	I13	I23		H51	H51	
		H51	H51	H51					H51	H51					H51	H51	
		H51	H51	H51					H51	H51					H51	H51	
G21	G31	G41	G31	G21	G31	G41	G31	G21	G31	G31	G41	G31	G21	G31	G41	G31	G21
G41	G21	G31	G31	G41	G21	G31	G41	G21	G31	G41	G21	G31	G41	G21	G31	G41	G21

ニットピッカーvs.ファイターフライのプログラムリスト

```
10 CLS:VIEW:CGEN 2:CGSET 1,2
:SPRITE ON:DIM X(3),Y(3),W(3
):LOCATE 10,21:PRINT"SCORE="
:PALET S 0,17,48,38,13 RETURN
20 DEF SPRITE 0,(1,1,0,0,0)=
"89:;":DEF SPRITE 1,(1,1,0,0
,0)="<=>?":FOR I=2 TO 6 STEP
 2:DEF SPRITE I,(0,1,0,0,0)=
CHR$(200)+CHR$(201)+CHR$(202
)+CHR$(203):DEF SPRITE I+1,(
0,1,0,0,0)=CHR$(204)+CHR$(20
5)+CHR$(206)+CHR$(207):NEXT RETURN
30 FOR I=1 TO 3:X(I)=0:Y(I)=
I*64:W(I)=0:NEXT:S=0:X=50:Y=
0:W=0 RETURN
40 FOR I=0 TO 7:SPRITE I:NEX
T RETURN
50 K=STICK(0):IF K<>0 THEN P
LAY"T1V15O1F1" RETURN
60 X=X+((K=2)-(K=1)-(X<16)+(
X>232))*8:Y=Y+W:FOR I=1 TO 3
 RETURN
70 IF Y(I)>140 THEN Y(I)=140
:W(I)=-22 RETURN
80 W(I)=W(I)+2 RETURN
90 Y(I)=Y(I)+W(I):X(I)=X(I)-
I*4:IF X(I)<0 THEN X(I)=240:
S=S+10:LOCATE 15,21:PRINT S RETURN
100 SPRITE I*2+F,X(I),Y(I):S
PRITE I*2+(F=1)+1:F=(F=1)+1:
NEXT RETURN
```

```
110 SPRITE (F=1)+1:SPRITE F,
X,Y:IF Y>140 THEN Y=140:W=-2
4 RETURN
120 W=W+2 RETURN
130 FOR I=1 TO 3:IF ABS(X(I)
-X)<16 AND ABS(Y(I)-Y)<16 TH
EN GOTO 150 RETURN
140 NEXT:GOTO 50 RETURN
150 LOCATE 9,23:PRINT"TRY AG
AIN ?";:PLAY"T200D3R3D6" RETURN
160 HS=PEEK(&H783A)*256+PEEK
(&H783B):IF HS<S THEN HS=S RETURN
170 POKE &H783A,HS/256,HS MO
D 256:LOCATE 9,22:PRINT"HI-S
CORE:";HS RETURN
180 IF STRIG(0)<>1 THEN 180 RETURN
190 RUN RETURN
```

●原作・K・K・Pの倉方俊輔『EAT EAT』(プログラムポシェットVOL・4)より移植

ゲーム7

敵はクルクルまわるスピナーだ！

スターシップ・ウォーズ

あそびかた　スピナーのたまは8方向に！

スターシップとスピナーの戦い(たたか)だ。キミは、✚ボタンでスピナーを上下左右(じょうげさゆう)にあやつり、チョコマカ動(うご)きまわるスピナーをうちおとそう。ミサイル発射(はっしゃ)はAボタンだ。

スピナーを1機(き)やっつけると、スコアが10点(てん)ふえて、またスピナーが現(あら)れる。スピナーにぶつからないようにご用心(ようじん)！

スター・シップ・ウォーズのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0			G52							
1							G53		G53	
2										
3		G50				G70				
4									G50	
5						G70				
6			G60						G53	
7								G53		
8		G70			G53	G53				
9				G52						
10						G61			G62	
11										
12								G63		
13					G72					
14										
15		G53		G52						
16								B10	C10	
17							F40	F10	F10	F60
18					G50			D30	A50	
19	G53									
20					G62					G63

プログラムはこんなふうにできているよ！

●変数リスト

A( ),B( )……移動用ワーク

HI……ハイスコア

X,Y……シップの座標

V……シップの方向

D,Q……スピナーの座標

W……スピナーの方向

M,N……シップの弾およびスピナーの座標（発射位置）

SC……得点

●プログラムの説明

10～ 40	初期設定
50	爆発のスプライト定義
60～ 70	シップ・スピナー・得点の表示
80～100	衝突判断
110～150	キー入力
160～170	弾がとまっていれば消去
180	シップの弾発射
190	シップ移動
200	スピナー移動
220～250	シップ移動サブルーチン
260～290	スピナー移動サブルーチン
300～360	スピナーの弾発射サブルーチン
370～390	スピナーの弾発射
400～410	スピナー撃墜・得点表示
420～460	ゲームオーバー・リプレー

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
			G52							G60							
		G60				G53					I60	J70	I60	I70			G60
	G60										J10	D40	D40	D70			
									G51		J40	D40	D40	D70			
											J40	J10	J60	J60			G50
G60			G50		G53												
								G61									
								G51									
	G53								G52						G61		
					G50							G70				G51	
			G60													G53	
						G50											
										G53							
G50					G52	G50								G60			
G50		G50														G51	
			G60				G52	G52			G52		G60				
															G60		
					G61											G50	
				G61					G72						G60		
		G62										G62					

スターシップ・ウォーズのプログラムリスト

```
10 VIEW:CGEN 2:CGSET 1,1:SPR
ITE ON:DIM A(3),B(3) RETURN
20 A(0)=4:A(2)=4:B(1)=4:B(3)
=4 RETURN
30 HI=PEEK(&H783A)*100+PEEK(
&H783B) RETURN
40 LOCATE 1,22:PRINT"HI ";HI
RETURN
50 DEF SPRITE 0,(3,1,0,0,0)=
CHR$(180)+CHR$(181)+CHR$(182
)+CHR$(183) RETURN
60 X=120:Y=120:V=1:GOTO 410
70 SPRITE 0:P=RND(209)+16:Q=
RND(153)+24:W=RND(4)*2+1:GOS
UB 230:GOSUN 270 RETURN
80 IF ABS(XPOS(1)-XPOS(2))<1
6 AND ABS(YPOS(1)-YPOS(2)).<1
6 THEN 420 RETURN
90 IF ABS(XPOS(1)-XPOS(4)+A(
(W-1)/2))<8 AND ABS(YPOS(1)-
YPOS(4)+B((W-1)/2))<8 AND MO
VE(4)<>0 THEN 420 RETURN
100 IF ABS(XPOS(3)-A((V-1)/2
)-XPOS(2))<8 AND ABS(YPOS(3)
-B((V-1)/2)-YPOS(2))<8 AND M
OVE(3)<>0 THEN 400 RETURN
110 S=STICK(0):T=STRIG(0) RETURN
120 IF S=8 THEN V=1 RETURN
130 IF S=1 THEN V=3 RETURN
140 IF S=4 THEN V=5 RETURN
150 IF S=2 THEN V=7 RETURN
160 IF MOVE(3)=0 THEN ERA 3 RETURN
170 IF MOVE(4)=0 THEN ERA 4:
GOSUB 350 RETURN
180 IF T=8 AND MOVE(3)=0 THE
N GOSUB 300 RETURN
190 IF MOVE(1)=0 THEN GOSUB
220 RETURN
200 IF MOVE(2)=0 THEN GOSUB
260 RETURN
210 GOTO 80 RETURN
220 X=XPOS(1):Y=YPOS(1) RETURN
230 DEF MOVE(1)=SPRITE(9,V,1
,4,0,0) RETURN
240 POSITION 1,X,Y:MOVE 1 RETURN
250 RETURN RETURN
```

```
260 P=XPOS(2):Q=YPOS(2):W=RN
D(4)*2+1 RETURN
270 DEF MOVE(2)=SPRITE(7,W,1
,4,0,1) RETURN
280 POSITION 2,P,Q:MOVE 2 RETURN
290 RETURN RETURN
300 M=XPOS(1)+A((V-1)/2):N=Y
POS(1)+B((V-1)/2) RETURN
310 IF M<0 OR M>240 OR N<5 O
R N>220 THEN RETURN RETURN
320 DEF MOVE(3)=SPRITE(12,V,
1,70,0,0) RETURN
330 POSITION 3,M,N:MOVE 3 RETURN
340 BEEP:RETURN RETURN
350 M=XPOS(2)+A((W-1)/2):N=Y
POS(2)+B((W-1)/2) RETURN
360 IF M<0 OR M>240 OR N<5 O
R N>220 THEN RETURN RETURN
370 DEF MOVE(4)=SPRITE(12,RN
D(7)+1,1,70,0,2) RETURN
380 POSITION 4,M,N:MOVE 4 RETURN
390 RETURN RETURN
400 SPRITE 0,XPOS(2),YPOS(2)
:ERA 2,3:PLAY"O3E1BE":SC=SC+
10 RETURN
410 LOCATE 15,22:PRINT"SCORE
";SC:GOTO 70 RETURN
420 SPRITE 0,XPOS(1),YPOS(1)
:ERA 1,4:PLAY"O1D1CE5":LOCAT
E 8,12:PRINT"GAME OVER" RETURN
430 IF HI<SC THEN POKE &H783
A,SC/100,SC MOD 100 RETURN
440 LOCATE 8,14:PRINT"REPLAY
?" RETURN
450 TR=STRIG(0):IF TR<>1 THE
N 450 RETURN
460 RUN RETURN
```

●原作・高橋司『グラスホッパーVS・キャノンボール』(プログラムポシェットVOL・4)より移植

ゲーム8

ひとりでもエキサイト！
スカッシュ・ゲーム

あそびかた　ボールがだんだんはやくなるぞ

キミの運動神経(うんどうしんけい)がものをいう、スポーツゲームだ。

かべにあたってはねかえってくるボールを、うまくラケットで打(う)ちかえそう。打(う)ちかえすたびに、スコアがあがっていくよ。

ラケットにあたらずに右(みぎ)のほうへボールが消(き)えたら、1回(かい)ミス。3回(かい)ミスっちゃうと、ゲームオー

スカッシュ・ゲームのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0										
1										
2	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50
3	A00	F10	M50	M60	M60	M60	M60	M60	M60	M60
4	A00	F10	M50	F50	F60	M60	M60	M60	M60	M60
5	A00	F10	M50	F50	F50	M60	M60	M60	M60	F50
6	A00	F10	M50	F50	F50	F50	M60	M60	F50	F50
7	A00	F10	M50	F50	F50	F50	F50	F50	F50	F50
8	A00	A10	M50	F50	F50	F50	F50	F50	F50	F50
9	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
10	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
11	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
12	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
13	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
14	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
15	A00	A10	M70	F50	F50	F50	F50	F50	F50	F50
16	A00	A10	M70	F50	F50	F50	M60	M60	F50	F50
17	A00	A10	M70	F50	F50	M60	M60	M60	M60	F50
18	A00	A10	M70	F50	F60	M60	M60	M60	M60	M60
19	A00	A10	M70	M60	M60	M60	M60	M60	M60	M60
20	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50

プログラムはこんなふうにできているよ！

●変数リスト

BX……ボールのX座標

BY……ボールのY座標

VX……ボールのX方向の移動量

VY……ボールのY方向の移動量

RX……ラケットのX座標

RY……ラケットのY座標

SC……スコア

●プログラムの説明

10～ 70	初期設定
80～180	ボールの位置によって動かし方を変える
190	ラケットにあたったか？
200～220	得点表示・ボールをはねかえす
230～300	ラケットの移動
320～380	ボールの移動
390～420	ゲームオーバー

バーだ。

最初はボールの動きもゆっくりだからかんたんだけど、打ちかえしているうちに、だんだんスピードがあがっていく。キミの運動神経はこのスピードにどこまでついていけるかな？

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60
M60	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
F50	F50																
M60	F50																
F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60

スカッシュ・ゲームのプログラムリスト

```
10 CLS:VIEW:CGEN 3:SPRITE ON
 RETURN
20 DEF SPRITE 0,(0,0,0,0,0)=
CHR$(207):DEF SPRITE 1,(0,0,
0,0,0)=CHR$(7):DEF SPRITE 2,
(0,0,0,0,0)=CHR$(7) RETURN
30 DEF SPRITE 3,(0,0,0,0,0)=
CHR$(207) RETURN
40 DEF SPRITE 4,(0,0,0,0,0)=
CHR$(207):PALET B 0,41,48,33
,2 RETURN
50 RX=220:RY=100:SX=220:SY=1
08:BX=RND(80)+70:BY=100:VX=2
:VY=2:K= RETURN
60 SPRITE 3,190,20:SPRITE 4,
200,20 RETURN
70 SPRITE 1,RX,RY:SPRITE 2,S
X,SY:LOCATE 4,1:PRINT"SCORE"
:LOCATE 10,1:PRINT SC RETURN
80 BX=BX+VX:BY=BY+VY RETURN
90 IF BX<35 THEN BX=34 RETURN
100 IF BX>220 AND BY<RY+14 A
ND BY>RY-4 THEN BX=220:GOTO
120 RETURN
110 IF BX>250 THEN BX=250 RETURN
120 SPRITE 0,BX,BY RETURN
130 IF BX>35 AND BX<216 THEN
 GOTO 170 RETURN
140 IF BX>=216 THEN GOTO 190
 RETURN
150 IF BY>184 OR BY<40 THEN
VY=-VY:BEEP RETURN
160 VX=-VX:BEEP:GOTO 230 RETURN
170 IF BY>184 OR BY<40 THEN
VY=-VY:BEEP RETURN
180 GOTO 230 RETURN
190 IF BY>RY+14 OR BY<RY-4 O
R BX<RX-8 OR BX>RX+8 THEN GO
TO 210 RETURN
200 SC=SC+20:LOCATE 10,1:PRI
NT SC:BEEP:GOTO 220 RETURN
210 GOTO 320 RETURN
220 VX=-VX-1 RETURN
230 FOR I=1 TO 10 RETURN
240 S=STICK(0) RETURN
```

```
250 IF S=8 THEN RY=RY-1:SY=S
Y-1 [RETURN]
260 IF S=4 THEN RY=RY+1:SY=S
Y+1 [RETURN]
270 IF RY<20 THEN RY=20:SY=2
8 [RETURN]
280 IF RY>212 THEN RY=210:SY
=218 [RETURN]
290 SPRITE 1,RX,RY:SPRITE 2,
SX,SY [RETURN]
300 NEXT [RETURN]
310 GOTO 80 [RETURN]
320 BX=BX+VX:BY=BY+VY [RETURN]
330 IF BX>240 OR BY>210 OR B
Y<10 THEN GOTO 360 [RETURN]
340 SPRITE 0,BX,BY [RETURN]
350 GOTO 230 [RETURN]
360 SPRITE 0:BEEP [RETURN]
370 IF K=1 THEN SPRITE 3:K=2
:FOR I=0 TO 200:NEXT:BY=RND(
80)+70:BX=40:VX=5:GOTO 80 [RETURN]
380 IF K=2 THEN SPRITE 4:K=3
:FOR I=0 TO 200:NEXT:BY=RND(
80)+70:BX=40:VX=5:GOTO 80 [RETURN]
390 LOCATE 8,10:PRINT"GAME O
VER" [RETURN]
400 LOCATE 8,14:PRINT"SCORE
IS";SC:LOCATE 4,1:PRINT"
                ":LOCATE 8,16:
PRINT"TRY AGAIN?" [RETURN]
410 IF STRIG(0)=1 THEN RUN [RETURN]
420 GOTO 410 [RETURN]
```

●原作・宮本智幸『テニス〃モドキ』(プログラムポシェットVOL・2)より移植

愛する地球をまもりぬけ！

ニタニタ・インベーダー

あそびかた　ニタニタの点滅に注意しよう

キミの住(す)む町(まち)へ、ニタニタ・インベーダーが攻(せ)めてきた。キミは町(まち)を守(まも)るために、ミサイル発射台(はっしゃだい)に乗(の)りこみ、敵(てき)に立(た)ち向(む)かった。

＋ボタンで発射台(はっしゃだい)を左右(さゆう)に動(うご)かし、Aボタンでミサイル発射(はっしゃ)だ。ニタニタは、少(すこ)しずつ、少(すこ)しずつキミのほうへ迫(せま)ってくるぞ。ニタニタに地上(ちじょう)を占領(せんりょう)されてしまった

ニタニタ・インベーダーのBG-GRAPHICデータ

	Ø	1	2	3	4	5	6	7	8	9
Ø	G12	G22								
1				G12	G22					
2								G70		
3										
4										
5										
6										
7										
8					G70					
9										
10										
11										
12									M60	M60
13										M60
14										
15										
16							G02	G02	G12	
17						G02	G22	G22	G22	G12
18					G02	G22	G22	G22	G22	G22
19				G02	G22	G22	G22	G22	G22	G22
20			G02	G22	G22	G22	G22	G22	G22	G22

プログラムはこんなふうにできているよ！

●変数リスト

FX……ファイターのX座標
FY……ファイターのY座標
IX……左のインベーダーのX座標
IY……左のインベーダーのY座標
C1,C3……レーザーのX座標
C2,C4……レーザーのY座標
VX……インベーダーの速度

●プログラムの説明

10～ 90　初期設定
100～170　インベーダーの移動
180～240　キー入力（ファイター移動用）
250～290　キー入力（レーザー１の発射用）
300～330　レーザー命中判断
340～370　キー入力（レーザー２の発明用）射
380～400　インベーダーを３基やっつけたときの処理
410～450　ゲームオーバー

ら、ゲームオーバーだ。

ときどき、ニタニタ側のスピナーが爆弾をまきちらしながら上空を横切っていく。この爆弾にも十分注意して戦ってくれ。

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
											G21						
G70									G01	G00	G10						
								G00	G20	G20	G20	G20	G10				
							G00	G20	G20	G20	G20	G20	G20	G10			
					G00	G20	G20	G20	G20	G20	G20	G20	G20	G20	G10		
				G00	G20	G20	G20	G20	G20	G20	G20	G20	G20	G10	G20	G10	
M60	M60	M60	M60	M60	M60												
M60	M60	M60	M60	M60	M60	M60											
M60	M60	M60	M60	M60	M60	M60	M60										
	M60	M60	M60	M60	M60	M60	M60	M60									
		M60	M60	M60	M60	M60	M60	M60	M60								
					M60	M60	M60		M60								
G22	G12																
G22	G22	G22	G12														
G22	G22	G22	G22	G22	G22	G22	G22	G12									

ニタニタ・インベーダーのプログラムリスト

```
10 FOR I=6 TO 7:DEF MOVE(I)=
SPRITE(12,1,2,100,0):NEXT [RETURN]
20 FOR I=0 TO 2:DEF SPRITE I
,(0,1,0,0,0)=CHR$(88)+CHR$(8
9)+CHR$(90)+CHR$(91):NEXT [RETURN]
30 FOR I=3 TO 5:DEF SPRITE I
,(0,1,0,0,0)=CHR$(92)+CHR$(9
3)+CHR$(94)+CHR$(95):NEXT [RETURN]
40 POSITION 6,100,220:POSITI
ON 7,100,220:SPRITE ON:VIEW:
DIM A(7):HS=PEEK(&H783A)*256
+PEEK(&H783B) [RETURN]
50 LOCATE 13,0:PRINT"HI-SCOR
E";HS [RETURN]
60 FOR I=0 TO 2:A(I)=I:A(I+3
)=I:NEXT [RETURN]
70 IX=50:IY=50:VX=3:F=1:K=0
80 DEF SPRITE 6,(0,1,0,0,0)=
CHR$(172)+CHR$(173)+CHR$(174
)+CHR$(175) [RETURN]
90 FX=100:FY=200:SPRITE 6,FX
,200 [RETURN]
100 IX=(IX+VX) MOD 256 [RETURN]
110 IF IX>180 AND VX>0 THEN
VX=VX+3:VX=-VX:IY=IY+3:GOTO
150 [RETURN]
120 IF IX<30 AND VX>0 THEN G
OTO 150 [RETURN]
130 IF IX>180 AND VX<0 THEN
GOTO 150 [RETURN]
140 IF IX<30 AND VX<0 THEN V
X=-VX:IY=IY+20:GOTO 150 [RETURN]
150 IF T=1 THEN GOTO 170 [RETURN]
160 SPRITE 0:SPRITE 1:SPRITE
 2:SPRITE A(3),IX,IY:SPRITE
A(4),IX+21,IY:SPRITE A(5),IX
+42,IY:T=1:GOTO 180 [RETURN]
170 SPRITE 3:SPRITE 4:SPRITE
 5:SPRITE A(2),IX+42,IY:SPRI
TE A(1),IX+21,IY:SPRITE A(0)
,IX,IY:T=0:GOTO 180 [RETURN]
180 PLAY"T1M1Y2O1C1":FOR I=1
 TO 10:S=STICK(0) [RETURN]
190 IF S=1 THEN FX=FX+1 [RETURN]
200 IF S=2 THEN FX=FX-1 [RETURN]
210 IF FX<40 THEN FX=40 [RETURN]
```

```
220 IF FX>200 THEN FX=200 [RETURN]
230 SPRITE 6, FX, FY [RETURN]
240 NEXT [RETURN]
250 S=STRIG(Ø) [RETURN]
260 IF S=8 AND F=1 AND MOVE(
6)=Ø THEN POSITION 6, FX, FY:F
=2:MOVE 6 [RETURN]
270 C1=XPOS(6):C2=YPOS(6):C3
=XPOS(7):C4=YPOS(7) [RETURN]
280 IF IX>C1 OR IX+58<C1 OR
IY>C2 OR IY+16<C2 THEN 310 [RETURN]
290 IF A(ABS(C1-IX)/19)=7 TH
EN 310 [RETURN]
300 BEEP:SC=SC+50:A(ABS(C1-I
X)/19)=7:A(ABS(C1-IX)/19+3)=
7 [RETURN]
310 IF IX>C3 OR IX+58<C3 OR
IY>C4 OR IY+16<C4 THEN 340 [RETURN]
320 IF A(ABS(C3-IX)/19)=7 TH
EN 340 [RETURN]
330 BEEP:SC=SC+50:A(ABS(C3-I
X)/19)=7:A(ABS(C3-IX)/19+3)=
7 [RETURN]
340 S=STRIG(Ø) [RETURN]
350 IF S=8 AND F=2 AND MOVE(
7)=Ø THEN POSITION 7, FX, FY:F
=1:MOVE 7 [RETURN]
360 FOR I=Ø TO 7:S=MOVE(I):I
F S=Ø THEN ERA I:POSITION I,
Ø, Ø [RETURN]
370 NEXT [RETURN]
380 IF A(Ø)=7 AND A(1)=7 AND
 A(2)=7 THEN SC=SC+5ØØ:LOCAT
E 1, Ø:PRINT"SCORE";SC:GOTO 6
Ø [RETURN]
390 LOCATE 1, Ø:PRINT"SCORE";
SC [RETURN]
400 IF IY<2ØØ THEN GOTO 1ØØ [RETURN]
410 LOCATE 8, 1Ø:PRINT"GAME O
VER":LOCATE 8, 14:PRINT"SCORE
 IS ";SC [RETURN]
420 IF HS<SC THEN HS=SC:POKE
 &H783A, HS/256, HS MOD 256 [RETURN]
430 LOCATE 8, 16:PRINT"TRY AG
AIN ?" [RETURN]
440 IF STRIG(Ø)<>1 THEN 43Ø [RETURN]
450 RUN [RETURN]
```

●原作・FP友の会会員『ぷんぺぷだあもどき』(プログラムポシエットVOL・2)より移植

ゲーム10

ピョンピョンはねてハイスコア！ ホップ・ボール

あそびかた リンゴのほうが点が高いぞ

右側にズラズラッと出てくる旗やリンゴをニタニタが取っていくゲーム。こんなふうに、キャラクタが下からどんどん出てきて上に消えていくゲームのことをスクロールゲームというんだよ。

ニタニタは、なにもしないでいると自動的に右端から左端へ、そしてまた左端から右端へゆっくり

ホップ・ボールのBG-GRAPHICデータ

	0	1	2	3	4	5	6	7	8	9
0										
1										
2		G72				G72			G72	G72
3		G72				G72		G72		
4		G72				G72		G72		
5		G72	G72	G72	G72	G72		G72		
6		G72				G72		G72		
7		G72				G72		G72		
8		G72				G72	G72	G72	G72	G72
9										
10										
11										
12						G72	G72			
13					G72			G72		
14					G72			G72		
15					G72			G72		
16					G72	G72	G72	G72		
17					G72				G72	
18					G72				G72	
19					G72				G72	G72
20						G72	G72	G72		

プログラムはこんなふうにできているよ！

●変数リスト

BX……ニタニタのX座標

A$,B$,C$,D$,……

……ニタニタの右横キャラクタ

T$……旗

TZ$……リンゴ

●プログラムの説明

10～ 50　初期設定

60～ 70　オープニング

80～110　はじめの画面を作る

120　旗の表示

130～140　リンゴの表示

150～200　ニタニタの移動

220～260　旗またはリンゴをとった

かどうか　かどうか？

270～320　ゲームオーバー処理

330～400　他のキャラクタ移動

410～440　ニタニタの方向転換

移動(いどう)する。でも、旗(はた)もリンゴもないところにあたるとアウトになっちゃうよ。そこで、ニタニタが左(ひだり)へ向(む)かっているときにAボタンを押(お)せばすぐに右(みぎ)へもどっていくから、これでタイミングをとろう！

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
			G72	G72	G72												
G72		G72				G72											
G72	G72	G72				G72											
G72		G72				G72											
G72		G72	G72	G72	G72												
G72		G72															
		G72															
								G72			G72						
								G72			G72						
		G72	G72					G72			G72						
	G72			G72				G72			G72						
G72					G72			G72			G72						
G72	G72	G72	G72	G72	G72			G72			G72						
G72					G72			G72			G72						
G72					G72			G72			G72						
G72					G72			G72			G72						
G72					G72	G72	G72	G72	G72	G72	G72	G72	G72	G72			

ホップ・ボールのプログラムリスト

```
10 VIEW:CGSET 1, 2:PALET B 0,
26, 48, 39, 22:SPRITE ON:K=1 RETURN
20 DEF SPRITE 0, (2, 1, 0, 0, 0) =
CHR$ (92) +CHR$ (93) +CHR$ (90) +C
HR$ (91) :SPRITE 0, 125, 80:BY=3
0 RETURN
30 DEF SPRITE 1, (2, 1, 0, 0, 0) =
CHR$ (92) +CHR$ (93) +CHR$ (90) +C
HR$ (91) RETURN
40 DEF SPRITE 2, (2, 1, 0, 0, 0) =
CHR$ (92) +CHR$ (93) +CHR$ (90) +C
HR$ (91) RETURN
50 TX=20:T$=CHR$ (215) :T2$=CH
R$ (199) :TY=23:BX=150:FLAG=0:
Y=10 RETURN
60 LOCATE 8, 10:PRINT"  HOP B
ALL":LOCATE 10, 12:PRINT"HIT
START" RETURN
70 S=STRIG (0) :IF S<>1 THEN G
OTO 70 RETURN
80 FOR I=1 TO 23:LOCATE 26, I
:PRINT CHR$ (199) :NEXT RETURN
```

```
90 DEF MOVE (0) =SPRITE (3, 8, 1,
255, 2) :DEF MOVE (1) =SPRITE (13
, 3, 1, 255, 3) RETURN
100 DEF MOVE (2) =SPRITE (15, 7,
1, 255, 0) :DEF MOVE (3) =SPRITE (
14, 8, 1, 255, 2) RETURN
110 SPRITE 1, 30, 180:SPRITE 2
, 50, 180 RETURN
120 LOCATE 26, TY:A=RND (3) :B=
RND (10) :IF A=2 AND B=9 THEN
PRINT T2$ RETURN
130 IF A=2 AND B<>9 THEN PRI
NT T$ RETURN
140 PRINT RETURN
150 S=STRIG (0) :IF S=8 AND FL
AG=1 THEN FLAG=0 RETURN
160 FOR I=1 TO 20 RETURN
170 IF FLAG=0 THEN GOSUB 440
 RETURN
180 IF FLAG=1 THEN GOSUB 420
 RETURN
190 SPRITE 0, BX, 100:NEXT RETURN
200 IF BX<215 THEN GOTO 350 RETURN
210 BB= (BX-16) /8 RETURN
220 A$=SCR$ (BB, Y) :B$=SCR$ (BB
+1, Y) :C$=SCR$ (BB-1, Y) :D$=SCR
$ (BB, Y+1) RETURN
```

```
230 IF A$=CHR$(199) OR B$=CH
R$(199) OR C$=CHR$(199) OR D
$=CHR$(199) THEN GOTO 330 [RETURN]
240 IF A$=CHR$(215) OR B$=CH
R$(215) OR C$=CHR$(215) OR D
$=CHR$(215) THEN GOTO 340 [RETURN]
250 IF K=1 THEN PLAY"T2O2A5A
4A1A5O3C4O2B1B4A1A4#G1A7":SP
RITE 2:K=2:GOTO 400 [RETURN]
260 IF K=2 THEN PLAY"T2O2A5A
4A1A5O3C4O2B1B4A1A4#G1A7":SP
RITE 1:K=3:GOTO 400 [RETURN]
270 PLAY"T2O2CEGO3CO2GEC" [RETURN]
280 LOCATE 8,10:PRINT"GAME O
VER" [RETURN]
290 LOCATE 9,12:PRINT"SCORE
IS  ";SC [RETURN]
300 LOCATE 8,14:PRINT"TRY AG
AIN" [RETURN]
310 IF STRIG(0)<>1 THEN 310 [RETURN]
320 RUN [RETURN]
330 LOCATE BB-1,Y:PRINT"    "
:LOCATE BB-1,Y+1:PRINT"    "
:SC=SC+10:PLAY"T2EC":LOCATE
4,0:PRINT"SCORE ";SC:GOTO 35
0 [RETURN]
```

```
340 LOCATE BB-1,Y:PRINT"    "
:LOCATE BB-1,Y+1:PRINT"    "
:SC=SC+20:PLAY"T2CEGO3C":LOC
ATE 4,0:PRINT"SCORE ";SC [RETURN]
350 N=RND(20):IF N>4 THEN 40
0 [RETURN]
360 IF N=0 THEN POSITION 0,2
55,239:MOVE 0:GOTO 400 [RETURN]
370 IF N=1 THEN POSITION 0,2
55,239:MOVE 1:GOTO 400 [RETURN]
380 IF N=2 THEN POSITION 0,2
55,239:MOVE 2:GOTO 400 [RETURN]
390 IF N=3 THEN POSITION 0,2
55,239:MOVE 3:GOTO 400 [RETURN]
400 FOR I=0 TO 3:IF MOVE(I)=
0 THEN ERA I [RETURN]
410 NEXT:GOTO 120 [RETURN]
420 IF BX<25 THEN BX=30:FLAG
=0:RETURN [RETURN]
430 BX=BX-1:RETURN [RETURN]
440 IF BX>225 THEN BX=220:FL
AG=1:RETURN [RETURN]
450 BX=BX+1:RETURN [RETURN]
```

●原作・中西康之『HOP BALL』(プログラムポシェットVOL.1)より移植

この本の内容についての問合せは、往復ハガキか
返信用封筒(60円切手添付)を同封して、
〒101 東京都千代田区神田錦町3-22
小笠原ビル4F

テクノポリス編集室

ファミリーベーシック入門係まで。

電話の場合は、

☎03-295-4610

まで。なお、電話による問合せはできるだけ
月曜日〜金曜日の午後5時〜7時
の間にお願いします。

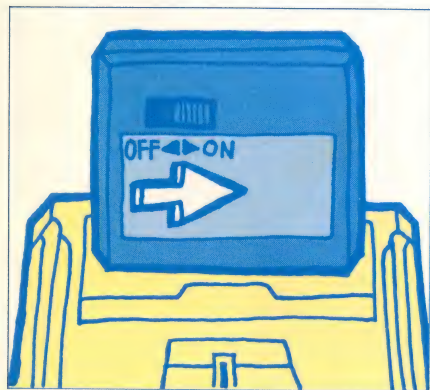
任天堂の**ファミリーベーシック入門**

編著者	テクノポリス編集部
発行者	栃窪宏男
発行所	株式会社徳間書店 〒105 東京都港区新橋4-10-1 ☎03-433-6231 振替　東京4-44392 テクノポリス編集部 〒105 東京都千代田区神田錦町3-22 小笠原ビル4F ☎03-295-4610
デザイン	ユートピア21／神田みき・富岡設子
イラスト	池田伊知郎
フィニッシュ	創文新社
印刷・製本	凸版印刷株式会社

ISBN4-19-723063-X

ファミリーベーシ

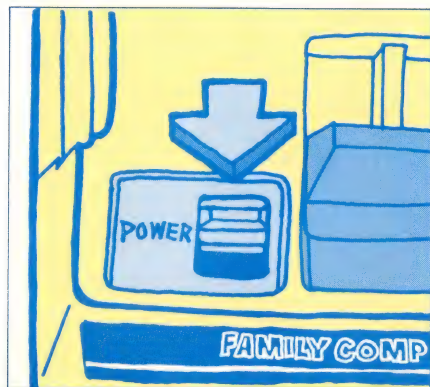
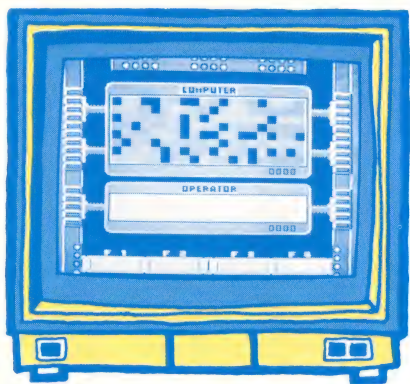
ファミリーベーシックのカセットは、バックアップスイッチを入れておくと、ファミリーコンピュータの電源スイッチを切ってもベーシックのプログラムを記憶しておくことができるんだ（もちろん電池を入れとかなきゃダメ)。でも操作中にやたらとバックアップスイッチをON・OFFすると、記憶がされなかったり、おかしな動作

①SYSTEM RETURNで、GAME BASICモード画面にもどる

③ファミリーベーシックのカセットのバックアップスイッチをONにする

ハックのしまい方

をすることがある。

スタート画面のときは、ファミコンがバックアップスイッチのON・OFFを指示してくれるから、それにしたがって操作しよう。でもBASICモードのときは指示が出ないから、下の手順でバックアップスイッチのOFFをしようね。

②「3……END」を選んで、スタートの画面にもどる

④ファミリーコンピュータの電源スイッチ(POWER)を切る

FAMILY BASIC™

ISBN4-19-723063-X C0076 ¥880E

任天堂の

ファミリー ベーシック入門

徳間書店 定価 880円

任天堂の
ファミリーベーシック入門

1985年 4 月30日　初版　　定価880円

編著者　テクノポリス編集部

発行者　栃窪宏男

発行所　株式会社徳間書店
〒105　東京都港区新橋4-10-1
☎03-433-6231
テクノポリス編集室
〒101　東京都千代田区神田錦町3-22
小笠原ビル4F　☎03-295-4610

印刷・製本　凸版印刷株式会社

落丁、乱丁がありましたときはお取りかえします。

この本を読んで

ファミリーベーシックの

プログラムをつくろう

キミ好みのゲームも

かんたんにできるよ！

任天堂のファミリーベーシック™入門

テクノポリス編集部：編著

徳間書店

任天堂の

ファミリーベーシック™入門

任天堂のファミリーベーシック™入門